Proceedings of the Fifth International Conference on Communication and Electronics Systems (ICCES 2020)
IEEE Conference Record # 48766; IEEE Xplore ISBN: 978-1-7281-5371-1

Virtual Mouse Using Object Tracking

Prof. Monali Shetty
Computer Engineering
Fr. CRCE, Mumbai.
shetty_monali@gmail.com

Christina A. Daniel
Computer Engineering
Fr.CRCE, Mumbai.
ca.daniel23@gmail.com

Manthan K.Bhatkar
Computer Engineering
Fr.CRCE, Mumbai.
manthanbhatkar@gmail.com

Ofrin P. Lopes
Computer Engineering
Fr.CRCE, Mumbai
lopesofrin28@gmail.com

Abstract—With new changes seen in computer technology day by day, it has become quite essential for us to find specific new ways of interaction with computer systems as its need is increasing in society every day. Today, every device is making the use of touch screen technology on its systems, which isn't affordable to be used in all applications. A specific interactive module like a virtual mouse that makes use of Object Tracking and Gestures that will help us to interact can be an alternative way for the traditional touch screen and the physical mouse. The objective is to create an Object Tracking application that interacts with the system. This system proposed is a Computer Vision-based mouse cursor control system, which uses hand gestures that are being captured from a webcam through an HSV color detection technique. This system allows the user to navigate the system cursor using their hand bearing color caps or tapes that the computer webcam tracks and perform mouse operations like left-click, right-click, and double click using different hand gestures. Python and OpenCV library is used for real-time computer vision to implement the system. The camera output is displayed on the monitor.

Key Words: Object Tracking, Virtual Mouse, Hand Gestures, Image capture, Image Pre-processing, Masking, Human-computer Interaction (HCI).

1. INTRODUCTION

One of the simplest, proficient, and significant ways of human communication is through hand gestures that people tend to make use of even unknowingly and a universally accepted language. In this paper, an online hand gesture mouse controlling system is proposed. In our setup of the system, a fixed integrated laptop camera is used to keep the cost at a minimum to implement the system in a cheap way affordable to be used by everyone without the need for any additional equipment that may incur a cost to the system and make it expensive, the camera system helps by capturing snapshots of the user object under a movement that is to be tracked using Red Green Blue [RGB] color space from a fixed distance and converted to HSV scheme for use by the system.

Under this project, a productive hand gesture segmentation technique has been recommended based on the pre-processing, background subtraction, and edge detection techniques [2]. Initially, before making use of the system, the color objects or color caps that will be used for tracking should be identified. Also, a threshold on the system should be set accordingly to detect these colored objects and store these thresholds for the system to make use of it while tracking. A range system has been initialized to set the HSV min, max values, and stored for use while tracking and performing operations. This is a significant part of the system as it may fail to detect objects and perform tracking if the range is not set correctly.

This system typically performs all operations that could be performed by a traditional mouse pointer. Operations like left-click, right-click, and hand recognition could easily perform other hand gesture operations like dragging a file to another location. The main aim is to create a cost-free hand recognition software for laptops and PCs with external webcams.

2. EXISTING SYSTEM

To solve the challenges faced in mouse detection and movement, a lot of research work is being carried out. N. Subhash Chandra et al. [1], & A. S.Ghotkar et al. [3] proposed systems using hand segmentation and skin color detection techniques. The methods used for tracking were reasonably straightforward, making the use of the webcam. Still, variation in human skin color complexion required the robust development of the algorithm for a natural interface, which made the recognition part quite tricky, and the results were not up to the mark. Convex Hull and Convexity defects were yet another method to detect hand and perform hand gestures shown by Sajjad Ur Rahman et al. [11]. The methods used for hand detection were quite good. Still, due to the skin detection and hand segmentation, the system faced another issue, i.e., it would detect other body parts seen on the screen and sometimes wouldn't detect accurately as skin color varies from person to person. Although these systems produce excellent results in detecting and tracking, the system is not quite good enough to make use of it in daily lives.

As shown in Table 1, Angel, Neethu.P.S [8] in her work proposed a Real-Time Static and Dynamic Hand Gestures Recognition where the hand tracking system had to be explicitly made adaptive for every user. This system was implemented in quite a restrictive indoor environment away

from where the environmental factors won't cause a problem. The system was quite prone to noise factors and a change in the light sensitivity of the illumination. J.L. Raheja et al [9] had proposed a system using the HSV algorithm, which used a special sensor Kinect to capture and process images; as a result, the user had to spend a lot of money for the sensor itself. Hence even today most people make use of the generic mouse or the trackpad system available on laptops. A system for accessing the screen pointer remotely using the hand gestures is unavailable. Though people are mostly trying to carry it out, the scope is merely restrictive in the field of a virtual mouse. Any further use of a hand tracking system has not been mainly made use of, as proposed by A. S.Ghotkar et al. [3]. They put forward a system that made the use of the static hand recognition system in which it merely did recognize the different shapes made by hand and defined individual action for each particular shape made, which was being limited. To remember different shapes to be prepared for different gestures resulting in large amounts of confusion and could be implemented with limited gestures.

Table 1. The methodology used and Accuracy Results of Existing Systems

Authors	Description	Recognition Method	Accuracy
J.L. Raheja , A.Chaudhary, K. Singal. [9]	The proposed algorithm uses a special sensor Kinect to capture and process images.	A Special Kinect Sensor had to be used by the user.	90-95%
N. Kim et al. [4][7]	It combines the structure-borne sound and the air-borne sound to recognize different types of hand grips.	SVM.	93%
Z. Xu, K. Wu, and P. Hu [5][7]	It uses gyroscope sensor auxiliary equipment and CFAR algorithm to reduce error recognition rate.	HMM, SVM.	95%
C. Yiallourides and P. P. Parada [6][7]	It uses two speakers to emit a sound signal with different frequency and controls the start and end of recording signals by a computer.	SVM	77.5%

Abhik Banerjee et al. [2] proposed a similar system that made use of RGB color bands on fingers user wears to provide information to the system. Once the system detects the colors, it would perform operations like tracking cursor and performing actions. The system was quite complex as it made use of different color bands for performing different actions, and actions were limited to mouse movement, left click and right click only. The system performed mouse movement on the detection of red color and would perform left-click if it detects red and green color. To navigate the system with ease was difficult for the user and wasn't used much.

3. PROPOSED SYSTEM

In the proposed system, the Object tracking method has been used to track the colored objects that help to operate on this system using the laptop webcam. By using the Object tracking system, the mouse and its basic operations like mouse pointing, selection, and deselection using left-click can be controlled. In a computer system, colors are represented in different formats like HSV (Hue Saturation Value) and BGR (Blue, Green, Red). With the BGR format, a pixel is represented by blue, green, and red parameters with blue being most significant and red being less significant. And each parameter of BGR usually having 0 – 255 values where 0 for all parameters represents black and 255 represents white, and the combinations of values for BGR from 0 to 255 creates various colors.

For example, a red pixel on your computer would have an R-value of 255, a B value of 0, and a G-value of 0. Your computer would interpret this as, “The pixel is 0 parts blue, 0 parts green, and 255 parts red.” HSV also represents pixels by 3 parameters but uses Hue, Saturation, and Value as parameters. HSV makes use of hue, which is the shade or color. The saturation is the intensity of the color. A value of 255 is the max intensity for saturation and 0 for saturation represents white. Saturation can also be known as how colorful a pixel is. How bright or dark a color is, is represented by its value. Since the HSV is used, it has been necessary to convert BGR to HSV which is done by OpenCV using the cv2.cvtColor( ) function and passing image and the flag as parameters that determines the image type of conversion to be done. An example of the flag for BGR to HSV is written as COLOR_BGR2HSV.

The proposed system uses Computer Vision libraries and algorithms to determine the object, its movements, and act as the movement using Real-time tracking. But our primary focus is on pointing the mouse and different actions by hand tracking and the movements performed by the hand and the fingers. The implemented system is written in Python language, which is quite easy to understand, with proper responsiveness, and is easy to implement. Hence Python is an

excellent language with great features and libraries that make the development of the system much easier, and it is a platform-independent, flexible, and portable language, which is desirable in creating such a program with the aim of for creating a virtual mouse using an object tracking system. This system is quite easy to use and also easy to grasp an understanding of how to perform different operations.

4. REQUIREMENTS FOR PROPOSED SYSTEM

The Virtual Mouse with Object Tracking uses nothing more than mere simple color caps or color clip on the fingers without any additional external requirements of the hardware for controlling the cursor to move and perform different operations. All this is being done with the help of the OpenCV computer vision library for tracking objects and gestures that simply makes it easy to work with Image Processing related work. OpenCV (Open Source Computer Vision) is a computer vision library of programming functions helping to solve computer vision problems. Simply said, it is an Image Processing library, mainly used to do different Image operations.

5. METHODOLOGY

In the Methodology, the procedure of each part is being explained step by step and their working will be shown in Fig.1.

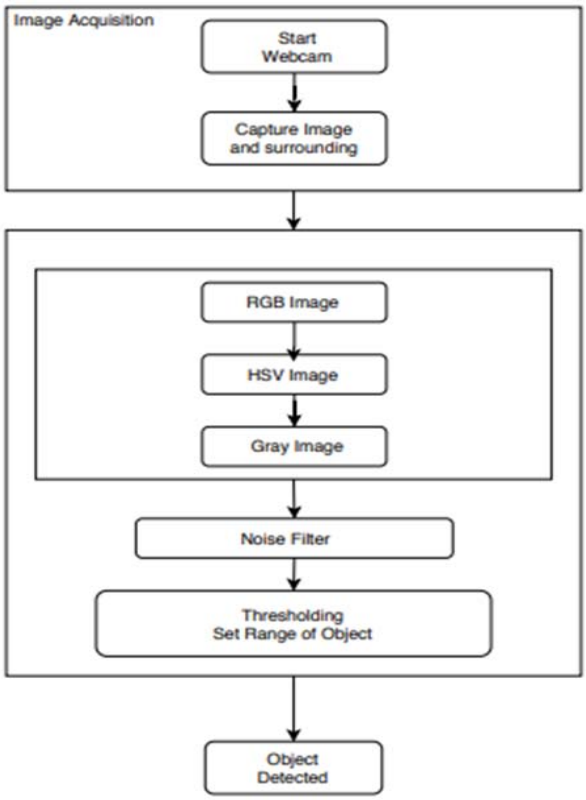

Fig 1: Design Flow Diagram

5.1 WebCam

The first important functioning hardware is a good working web camera. To capture the object used to instruct the mouse and replace the working of it with the object. So that once if the object is captured in our frames using OpenCV then it can be used as our mouse to make the functioning easier. The purpose of the Webcam is capturing the hand gesture and movements generated by humans and store its image in memory.

5.2.1 Collecting Frame (Detecting Object)

Once the web camera is working correctly, the project can be started i.e to install the project on a laptop. Once that is done the project will be started, by doing this the webcam will start automatically and a window will pop up showing your surroundings. This can be done using OpenCV Library as it can detect different colors in surroundings. To capture the object in your hand, the threshold has to be set i.e to convert RGB TO HSV. Different arguments should be set for range filter and the person using it adjusts the different HSV ranges and identifies the one specific object color and sets it for use.

For Example: As shown in Fig.2, if you are using any yellow color object, the work should be associated the HSV minimum and maximum threshold till only the yellow-colored object is being detected, this changes the object from RGB TO HSV format and set it for use i.e the object identified/detected will now work as your mouse.

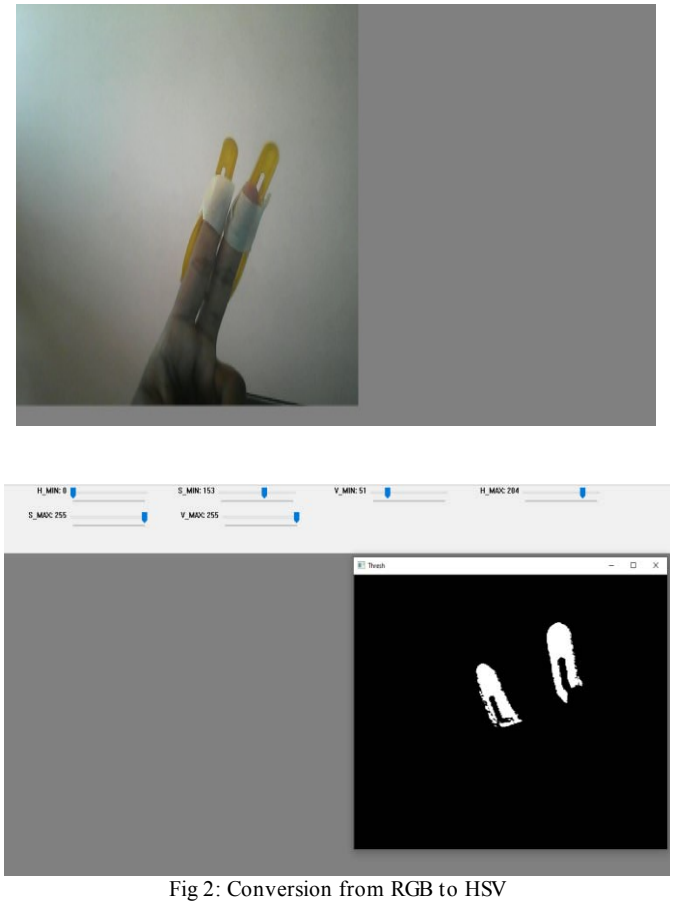

Fig 2: Conversion from RGB to HSV

5.2.2 Range Filter

A window of different range values from MIN to MAX is created i.e. range from 0-255. Once you start the range filter three window pops-up: 1. Original 2. Range-Setter 3. Threshold. So from the original frame, the threshold values of the object of which should be used as a mouse should be set. That is it should be converted from RGB to HSV as shown in the previous image. The image.copy( ) function is used to copy the frame image from original to threshold. Cam.read( ) to read the image from webcam.

5.3 Frame after setting the range

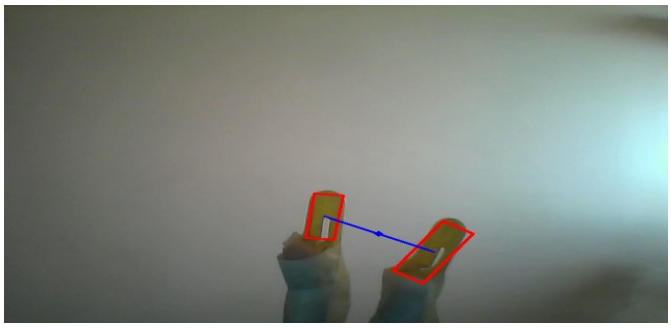

Fig 3: Object detection

The threshold value is set and the range is detected as shown in Fig 3, this starts our mouse tracking. When it has been done, it automatically starts the webcam and starts to search for items with the set thresholds and detects the object as shown above. Now this object will work as the mouse and it is performed for different functions. It should be made sure that the two objects are detected and kept in one frame if the connection of two objects is out of the frame it will not work as desired.

5.4 Noise

Noise is quite an essential factor in any computer vision project that needs to be addressed. As the system uses a webcam for capturing frames, there can be issues that arise due to the environment. All the system needs is the right lightning environment to capture object movements. To decrease noise in the system, it has to be worked out in the places where environmental factors like the sun won't cause glares or effect on capturing the images. Also, it should make use of the MedianBlur( ) filter to remove noise by some averaging operations and GaussianBlur( ) to remove high-frequency components so that system works better. As different systems are designed by different manufacturers, there will be a difference in terms of megapixels of the camera and there is no specific predefined range of filtering window that will work for all colors. Hence the easiest way to set the system for use is to take the colored object in front of a webcam and then slowly adjusting the bottom parameters(H MIN, V MIN, S MIN) one after the other and then start to slowly decrease the upper parameters (H MAX, V MAX, S MAX). When you are done adjusting you will find that the entire image will be dark and only the colored object has a corresponding white patch.

5.5 Mouse Movement

The movements of the object should be tracked using video capture in OpenCV and detect the objects, whether if they are together or separate to perform different operations. The screen size should be set to gui.size( ), and then divide it by a damping factor to reduce extra movement as the object in real-world moves a lot across the frame and its movement needs to be reduced according to screen ratio, else mouse will move all across the screen even when it is needed to move only a little bit, and then read the object using the cam.read() function, then capture the centers of two objects and set it. Finally, convert the detected coordinates from the camera to the actual screen resolution. After that, the location of the mouse_position has been set by performing some scaling operations. The data has been collected from range.pickle( ), the capture frame using videocapture( ), and collect points of upper and lower HSV then set height, width, and screen size, this can locate the two objects used and perform mouse movement i.e. going up, down, left, right.

5.6 Perform Different Operations

Operations are performed by capturing the movements in a frame. The various operations that can be performed are:

- **Right Click**

Keep the objects together for 10-20 frames then release.

- **Left Click**

Keep the objects together for less than 10 frames then release it.

- **Double Click**

Open using right-click or double click

Close using mouse movement

Real-Time Operation:

- Pause/Play any video :

If you are watching a video one can bring objects together and pause and play the video.

6. RESULT AND EVALUATION

The idea behind making this project was the ease of use for every individual. Replacement of the mouse with any object so that they can use this function for what they desire from anywhere concerning be in the frame of web camera. The use of this project in real-time is vast.

The system developed first takes over your mouse and the person using it can then move the object which will work as a mouse. Following operations can be performed by the

user click, right-click, left-click, select, pause play any video online using the object. The object is first detected by the color and as the threshold is set when the camera starts it automatically detects the object in hand and uses that as a mouse. It's a very sufficient way to use an object in any surrounding without the use of any specific external hardware or extra camera, the web camera in the laptop does it all. The flowchart in Fig 4 shows the complete process of how the system works from capturing an image through a webcam to processing individual frames, converting from RGB to HSV format, and saving it to see how it performs different operations

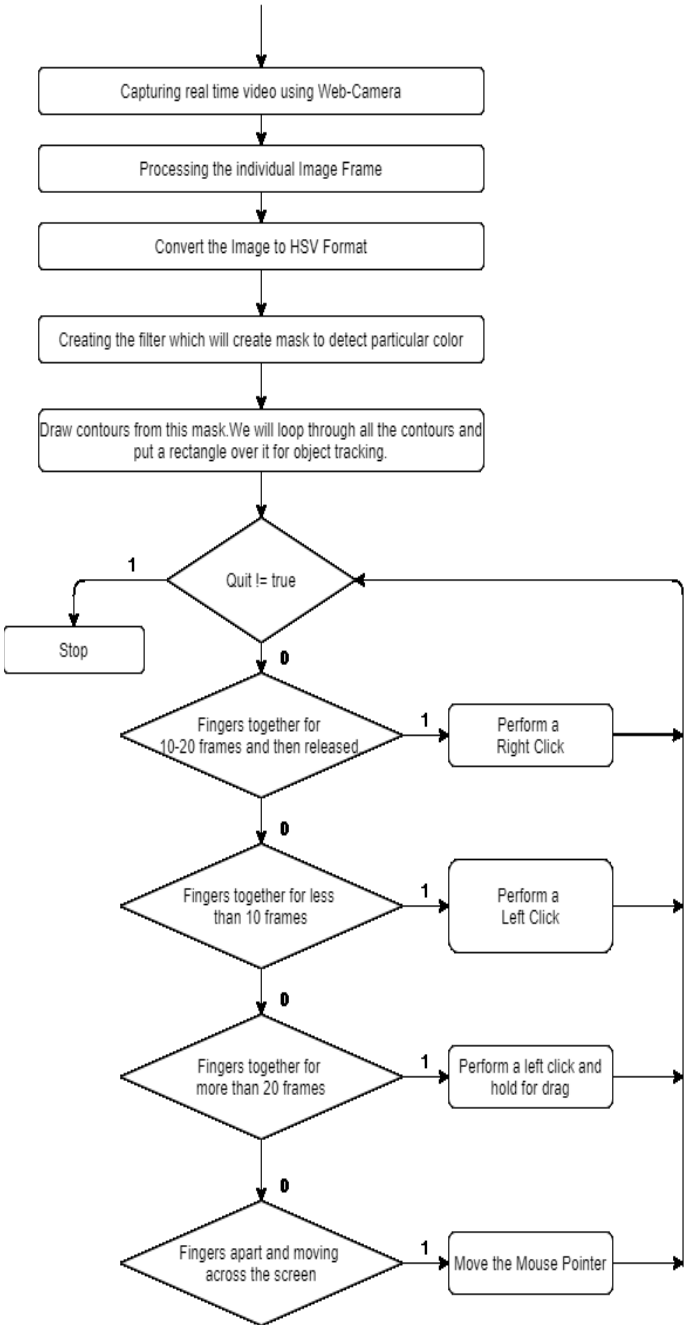

Fig 4: Flowchart

Our main aim of this system was to reduce any external hardware components needed. Although the application can be run in any computer with a webcam, ideally, it is recommended that at least a 2 MP front camera with at least a Core-i5 processor and at least 2GB RAM. Following in Fig 5 and 6 are the snapshots of the system in use.

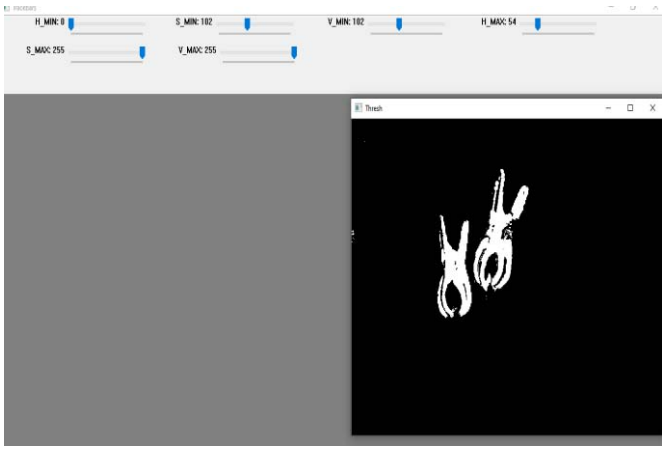

Fig 5: Setting a Threshold for Object

Fig 6: Mouse Movement

Table 2: Accuracy Results of Proposed System

Inputs	Mouse events	Accuracy with a plain background (in %)	Accuracy With non-plain Background (in %)
Two-color object (open Gesture)	Mouse Movement	95	40

Closer two-color object (Closed Gesture)	Left Button Click	89	41
Keep close for 5 seconds (Closed Gesture)	Double Click	87	42
Single color cap (open Gesture)	Right Button Click	96	80
Swipe up Down	Scrolling up or down	75	40

7. CONCLUSION

In this paper, a system has been proposed to recognize objects and perform the mouse function, which includes the movement on the mouse cursor, the drag, and click. Gesture Recognition tech has a vast variety of different applications in various fields of computer graphics, augmented reality systems, Virtual Reality gaming, and many more. The most popular use of gestures can be in the field of digital painting in VR, where people can paint in 3D systems where their work can be in 3D which is quite amazing. The artist would be able to create 2D, 3D images using object tracking technology using their hands as a brush and a monitor as a display set, and a virtual reality kit to perform work in the VR environment. In the case of gaming, many modern gaming consoles have this technology applied to them where a person can play interactive games by tracking motions and performing actions.

There are different tools for gesture recognition. A new Object tracking hand gesture recognition algorithm that overcomes the challenges for the detection of hand gesture images is developed in this paper. It works well compared with the existing system as shown in Table 1 even in the presence of background clutter as shown in Table 2.

Gesture recognition has been the best way for any interaction between machines and humans. For developing alternative human-computer interaction modules, gesture recognition is also essential. It has enabled us to interface with different computer systems more naturally. Gesture recognition can be deemed to be useful for many applications for people with disabilities.

8. REFERENCES

[1] N. Subhash Chandra, T. Venu, P. Srikanth, "A Real-Time Static & Dynamic Hand Gesture Recognition System", International Journal of Engineering Inventions, Volume 4, Issue 12, August 2015.

[2]Abhik Banerjee, Abhirup Ghosh, Koustuvmoni Bharadwaj," Mouse Control using a Web Camera based on Color Detection", IJCTT, vol.9, Mar 2014.

[3] A. S.Ghotkar and G. K. Kharate, " Hand Segmentation Techniques to Hand Gesture Recognition for Natural Human-Computer Interaction", International Journal of Human-Computer Interaction, vol. 3, no. 1, pp. 15-25, 2012.

[4] N. Kim and J. Lee, "Towards grip sensing for commodity smartphones through acoustic signature," In Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. ACM Int. Symp. Wearable Comput., Maui, HI, USA, 2017, pp. 105-108.

[5] Z. Xu, K. Wu, and P. Hu, "Ultrasonic waves based gesture recognition method for smartphone platform," Computer. Eng. Appl., vol. 54, no. 2, pp. 239-245, Jan. 2018. doi: 10.3778/j.issn.1002-8331.1608 0081.

[6]C. Yiallourides and P. P. Parada, ``Low power ultrasonic gesture recognition for mobile handsets," in *Proc.* IEEE Int. Conf. Acoust., Speech SignalProcess. (ICASSP), Brighton, U.K., May 2019, pp. 2697_2701.

[7] Zhengjie Wang , Yushan Hou , Kangkang Jiang , Wenwen Dou , Chengming Zhang , Zehua Huang , And Yinjing Guo ,"Hand Gesture Recognition Based on Active Ultrasonic Sensing of Smartphone: A Survey" , Digital Object Identifier 10.1109/ACCESS.2019.2933987

[8] Angel, Neethu.P.S, "Real-Time Static & Dynamic Hand Gesture Recognition", International Journal of Scientific & Engineering Research Volume 4, Issue3, March-2013.

[9] J.L. Raheja, A. Chaudhary, K. Singal, 2011, "Tracking of Fingertips and Centre of Palm using KINECT ", In proceedings of the 3rd IEEE International Conference on Computational Intelligence, Modelling and Simulation

[10] Monika Deswal, Neetu Sharma, "A Fast HSV Image Color and Texture Detection and Image Conversion Algorithm ", International Journal of Science and Research (IJSR), Volume 3 Issue 6, June 2014.

[11] Sajjad Ur Rahman, Zeenat Afroze, Mohammed Tareq. " Hand Gesture Recognition Techniques For Human Computer Interaction Using OpenCv", International Journal of Scientific and Research Publications, Volume 4, Issue 12, December 2014.

[12] Rashmi Adatkar, Ronak Joshi, Mehul Jani, Prashant Karn, "Virtual Mouse ", Imperial Journal of Interdisciplinary Research (IJIR), Vol-3, Issue-4, 2017.