A Mini Project Report
ON
VIRTUAL MOUSE PROJECT

A project report submitted in partial fulfillment of the Academic requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

By

N.Ajay	(17TQ1A0562)
A.Bhavana	(17TQ1A0508)
A.Sai Kumar	(17TQ1A0506)
K.Abhilash	(17TQ1A0546)
K.Shiva Kumar	(18TQ5A0506)

Under the guidance of
Dr.R.Dinesh Kumar M.E., Ph.D
Professor, Dept. of C.S.E
2020-2021

SIDDHARTHA INSTITUTE OF TECHNOLOGY AND SCIENCES
DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY
(Affiliated to JNTUH & O.U, Hyderabad. Recognized by AICTE)
Korremula Road, Narapally, Ghatkesar, Medchal-Malkajgiri 2020

CERTIFICATE

This is to certify that the project entitled "VIRTUAL MOUSE PROJECT" is a bonafide work carried out by

N.Ajay	(17TQ1A0562)
A.Bhavana	(17TQ1A0508)
A.Sai Kumar	(17TQ1A0506)
K.Abhilash	(17TQ1A0546)
K.Shiva Kumar	(18TQ5A0506)

In partial fulfilment of the requirement for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** from Siddhartha Engineering College, affiliated to JNTU, Hyderabad, under our guidance and supervision. The results presented in this project have been verified and are found to be satisfactory. The results embodied in this project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	Head of the department
Dr.r. Dinesh kumar	**Dr.Satyanarayana**
Assistant Professor	Professor &HOD
Department of CSE	Department of cse
SITS, Hyderabad	SITS, Hyderabad

External Examiner

DECLARATION

This is to certify that the work reported in the present project entitled **"VIRTUAL MOUSE PROJECT"** is a record of work done by us in the Department Computer Science and Engineering, Siddhartha Engineering College, JNTU Hyderabad . The reports are based on the project work done entirely by us and not copied from any other source. We submit our project for further development by any interested students who share similar interests to improve the project in the future.

The results embodied in this project report have not been submitted to any other university or Institute for any degree or diploma to the best of our knowledge and belief.

N. Ajay (17TQ1A0562)

A. Bhavana (17TQ1A0508)

A. Sai Kumar (17TQ1A0506)

K. Abhilash (17TQ1A0546)

K. Shiva Kumar (18TQ5A0506)

ACKNOWLEDGMENT

We are extremely grateful to **Dr. Battacharya.S Principal** and **Dr.Satyanarayana**, HOD, **Department of CSE,** Siddhartha Engineering College for their Constant support.

I am extremely thankful **to DR.R.Dinesh Kumar** , Assistant Professor, Internal Guide, Department of CSE, for his/her constant guidance , encouragement and moral support throughout the project.

I will be failing in duty if I do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Project.

I express my thanks to all staff members and friends for all the help and co-ordination extends to bringing out this project successfully in time.

Finally, I am very much thankful to my parents who guided me for every step.

N.Ajay (17TQ1A0562)

A.Bhavana (17TQ1A0508)

A.Sai Kumar (17TQ1A0506)

K.Abhilash (17TQ1A0546)

K.ShivaKumar (18TQ5A0506)

ABSTRACT

This document proposes a novel camera vision based cursor control system, using hand gestures captured from a webcam through a color detection technique. The system will allow the user to navigate the computer cursor using their hand bearing color caps or tapes and left click and dragging will be performed using different hand gestures. And also it performs file transfer between two systems in a single same network. The proposed system uses nothing more than a low resolution webcam that acts as a sensor and it is able to track the users hand bearing color caps in two dimensions. The system will be implemented using the python and OpenCV. The hand gesture is the most effortless and natural way of communication. The output of the camera will be displayed on the monitor. Shape and position information about the gesture will be gathered using detection of color. The file transferring scheme is implemented by using the python server programming.

***Key Words*: Virtual Mouse, Hand Gestures, Image capture, Processing, Masking, DnD Frame**

CONTENTS

TOPIC **Page No**

1.INTRODUCTION

The most efficient and expressive way of human communication is through hand gesture, which is a universally accepted language. It is pretty much expressive such that the dumb and deaf people could understand it. In this work, real-time hand gesture system is proposed. Experimental setup of the system uses fixed position low cost web camera high definition recording feature mounted on the top of monitor of computer or a fixed camera on a laptop, which captures snapshot using Red Green Blue [RGB] color space from fixed distance. This work is divided into four stages such as image preprocessing, region extraction, feature extraction, feature matching. Recognition and the interpretation of sign language is one of the major issues for the communication with dump and deaf people.

In this project an effective hand gesture segmentation technique has been proposed based on the preprocessing, background subtraction and edge detection techniques [1]. Pre-processing is defined as procedure of formulating data for another process. The main objective of the preprocessing process is to transform the data into a form that can be moreeffectively and effortlessly processed. In the proposed work, the pre-processing techniques are created on the basis of Different types of combinations from the subsequent hand gesture image processing operations such as capturing image, removing noise, background subtraction, and edge detection and these image processing methods are discussed is as follows.

Initially, the hand gesture images are captured from the vision based camera, The hand gestures can be observed with the different kind of interfaces like “data gloves” that accurately records every abduction angles and digit’s and position sensors for wrists and optical orientation or electromagnetic, requiring the user to wear trackers or gloves. Habitually, the glove based interfaces even need the user to be hitched to the computer, all minimizing time to user comfort and interface, conversely, vision based interfaces offer unencumbered interaction with human.

By this documentation we are aiming in creating a cost free hand recognition software for laptops and PCs with a web-cam support. The project covers as a hand recognition tool which could be used to move the mouse pointer, perform simple operations like clicking and other hand gesture operations like moving file from computer to computer through delicate socket programming and performing simple but fascinating operations that could be covered with the hand recognition.

2. EXISTING SYSTEM

The existing system consists of the generic mouse and trackpad system of monitor controlling and the non availability of a hand gesture system. The remote accessing of monitor screen using the hand gesture is unavailable. Even-though it is largely trying to implement the scope is simply restricted in the field of virtual mouse. The existing virtual mouse control system consists of the simple mouse operations using the hand recognition system, where we could perform the basic mouse operation like mouse pointer control, left click, right click, drag etc. The further use of the hand recognition is not been made use of. Even-though there are a number of systems which are used for hand recognition, the system they made used is the static hand recognition which is simply recognition of the shape made by hand and by defining an action for each shape made, which is limited to a number of defined actions and a large amount of confusion.

3. PROPOSED SYSTEM

Using the current system even-though there are a number of quick access methods available for the hand and mouse gesture for the laptops, using our project we could make use of the laptop or web-cam and by recognizing the hand gesture we could control mouse and perform basic operations like mouse pointer controlling, select and deselect using left click, and a quick access feature for file transfer between the systems connected via network LAN cable. The project done is a “Zero Cost” hand recognition system for laptops, which uses simple algorithms to determine the hand, hand movements and by assigning an action for each movement[2]. But we have mainly concentrated on the mouse pointing and clicking actions along with an action for the file transfer between connected systems by hand action and the movements. The system we are implementing which is been written in python code be much more responsive and is easily implemented since python is a simple language and is platform independent with a flexibility and is portable which is desirable in creating a program which is focused in such an aim for creating a Virtual Mouse and Hand Recognition system. The system be much more extendable by defining actions for the hand movement for doing a specific action. It could be further modified to any further extent by implementing such actions for the set of hand gestures, the scope is restricted by your imagination.

4. USE OF PROPOSED WORK :

This Virtual Mouse Hand Recognition application uses a simple color cap on the finger without the additional requirement of the hardware for the controlling of the cursor using simple gestures and hand control. This is done using vision based hand gesture recognition with inputs from a webcam.

5. METHODOLOGY :

In the Methodology, the method used in each component of the system will be explained separately. They are following subsections:

5.1 Camera Settings

The runtime operations are managed by the webcam of the connected laptop or desktop. To capture a video, we need to create a Video Capture object. Its argument can be either the device index or the name of a video file. Device index is just the number to specify which camera. Since we only use a single camera we pass it as '0'. We can add additional camera to the system and pass it as 1,2 and so on. After that, you can capture frame-by-frame. But at the end, don't forget to release the capture. We could also apply color detection techniques to any image by doing simple modifications in the code.

5.2 Capturing frames

The infinite loop is used so that the web camera captures the frames in every instance and is open during the entire course of the program. We capture the live feed stream, frame by frame. Then we process each captured frame which is in RGB(default) color space to HSV color space. There are more than 150 color-space conversion methods available in OpenCV. But we will look into only two which are most widely used ones, BGR to Gray and BGR to HSV.

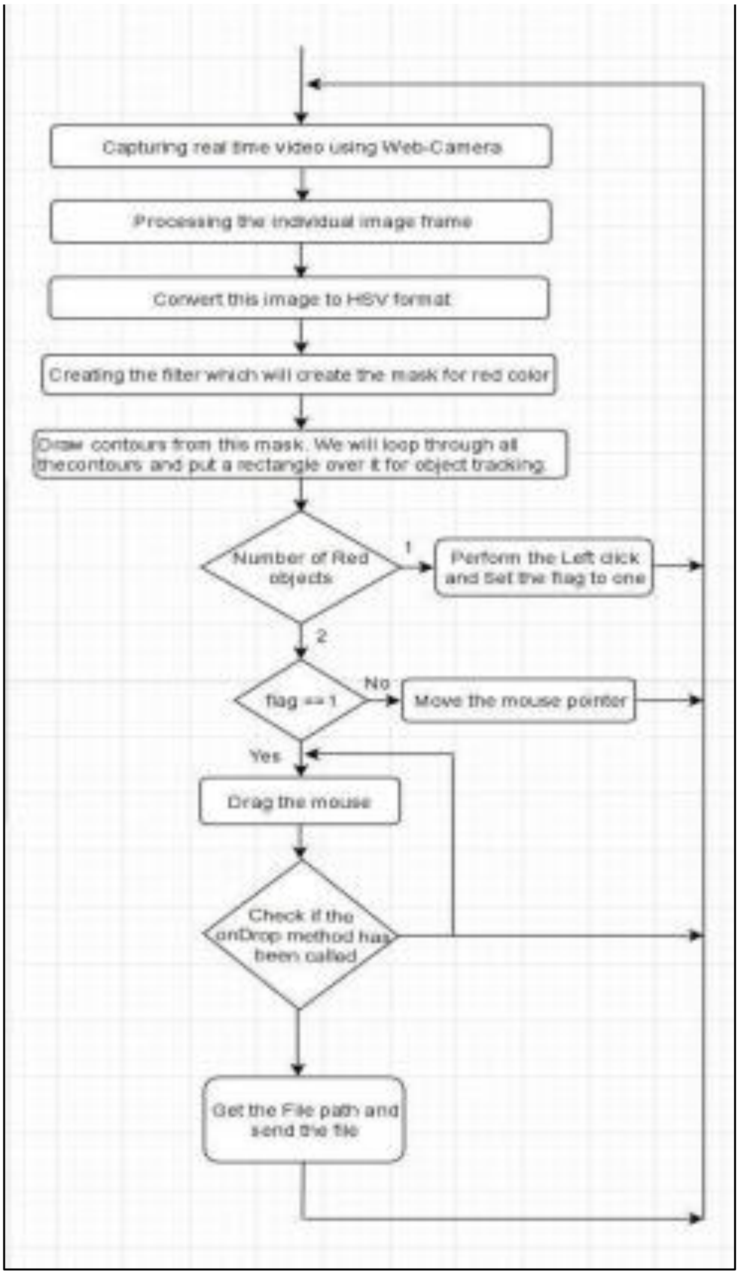

Capturing real time video using Web-Camera
Processing the individual image frame
Convert this image to HSV format
Creating the filter which will create the mask for red color
Draw contours from this mask. We will loop through all thecontours and put a rectangle over it for object tracking.
Number of Red objects
1
Perform the Left click and Set the flag to one
2
flag == 1
No
Move the mouse pointer
Yes
Drag the mouse
Check if the onDrop method has been called
Get the File path and send the file

Fig -1: Flowchart

5.3 Masking technique

The mask is basically creating some specific region of the image following certain rules. Here we are creating a mask that comprises of an object in red color. After that we perform a bitwise AND operation on the Input image and the Threshold image, which result in only the red colored objects are highlighted. This result of the AND operation is stored in res. We then display the frame, res and mask on 3 separate windows using imshow() function.

5.4 Display the frame

The imShow() is a function of HighGui and it is required to call the waitKey regulerly. The processing of the event loop of the imshow() function is done by calling waitKey. The function waitKey() waits for key event for a “delay” (here, 5 milliseconds). Windows events like redraw, resizing, input event etc. are processed by HighGui. So we call the waitKey function, even with a 1ms delay.

5.5 Mouse Movement

We have to first calculate the center of both detected red object which we can easily do by taking the average of the bounding boxes maximum and minimum points. now we got 2 co-ordinate from the center of the 2 objects we will find the average of that and we will get the red point shown in the image. We are converting the detected coordinate from camera resolution to the actual screen resolution. After that we set the location as the mouse_position. but to move the mouse pointer it will take time. So we have to wait till the mouse pointer reaches that point. So we started a loop and we are not doing anything there we are just waiting will the current mouse location is same as assigned mouse location. That is for the open gesture.

5.6 Clicking

The next step is to implement the close gesture. The operation is performed by clicking the object and dragging it. It is similar to the open gesture, but the difference is we only have one object here so we only need to calculate the center of it. And that will be placed on the location where we will position our mouse pointer. Instead of mouse release operation we will be performing a mouse press operation.

5.7 Drag

In order to implement the dragging we indroduce a variable ‘pinchflag’. It will be set to 1 if it was clicked earlier. So after clicking whenever we find the open gesture we check if the pinchflag is set to 1. If it is set to one then Drag operation is performed otherwise the mouse move operation is performed.

5.8 DnD Frame

First we create the MyFileDropTarget class. Inside that we have one overridden method, OnDropFiles. This method accepts the x/y position of the mouse along with the file paths that are dropped.

5.9 File Transfer

Using the file path obtained from MyFileDropTarget class we send the file to the other system using the python socket module. And there it recives and creates the file.

6. Application:

This project can be useful for presentations and for reducing workspace and burden of extra hardware devices. In critical events such as battleground, operation theater, this project will be able to withstand itself, as it is wispier than any other contemporary system in the market for computer interaction. Since it removes the burdens of devices, it brings the user and the workspace more closer than before.

Major applications:

• Robot controlling is one of the major applications of this system. Controlling robots without machines or extra devices can be an excellent addition to this technology.
• Digital artist can draw 2D or 3D images using this gesture-based mouse on digital canvases. It will allow more freedom more dynamic more space to artists to create their art.
• Critical events like a battleground, operation theater, mining fields can be controlled by gesture mouse.
• Virtual Reality or Augmented Reality based games can be played without any extended or wireless devices with bare hands more comfortably .
.• For patients with no limb control, this system can be proved very useful and effective.
• This mouse system can be applicable for sign language for the dumb & deaf people. It can help them to interact with computing machines.

7.Source code:

```
 import cv2 as cv

import numpy as np

from pynput.mouse import Button, Controller

import wx


mouse = Controller()

app = wx.App(False)

(sx, sy) = wx.GetDisplaySize()

(camx, camy) = (320, 240)


cam = cv.VideoCapture(0)

cam.set(3, camx)

cam.set(4, camy)


mlocold = np.array([0, 0])

mouseloc = np.array([0, 0])

damfac = 2.5

pinch_flag = 0

while (1):

    ret, img = cam.read()
    img = cv.GaussianBlur(img, (5, 5), 0)
    hsv_img = cv.cvtColor(img, cv.COLOR_BGR2HSV)
    mask = cv.inRange(hsv_img, np.array([33, 80, 40]), np.array([102, 255, 255]))

    mask_open = cv.morphologyEx(mask, cv.MORPH_OPEN, np.ones((5, 5)))
    mask_close = cv.morphologyEx(mask_open, cv.MORPH_CLOSE, np.ones((20, 20)))
    mask_final = mask_close

    conts, _ = cv.findContours(mask_final.copy(), cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)
    cv.drawContours(img, conts, -1, (255, 0, 0), 3)
    if (len(conts) == 2):
        if (pinch_flag == 1):
            pinch_flag = 0
            mouse.release(Button.left)
        x1, y1, w1, h1 = cv.boundingRect(conts[0])
        x2, y2, w2, h2 = cv.boundingRect(conts[1])
        cv.rectangle(img,(x1,y1),(x1+w1,y1+h1),(255,0,0),2)
        cv.rectangle(img,(x2,y2),(x2+w2,y2+h2),(255,0,0),2)
        cx1 = round(x1 + w1 / 2)
        cy1 = round(y1 + h1 / 2)

        cx2 = round(x2 + w2 / 2)

        cy2 = round(y2 + h2 / 2)

        cv.line(img, (cx1, cy1), (cx2, cy2), (255, 0, 0), 2)

        cx = round(cx1 / 2 + cx2 / 2)

        cy = round(cy1 / 2 + cy2 / 2)

        cv.circle(img, (cx, cy), 2, (0, 0, 255), 2)

        mouseloc = mlocold + ((cx, cy) - mlocold) / damfac

        mouse.position = (round(sx - (mouseloc[0] * sx / camx)), 
    round((mouseloc[1] * sy / camy)))

        mlocold = mouseloc


    elif (len(conts) == 1):

        if (pinch_flag == 0):

            pinch_flag = 1

            mouse.press(Button.left)


        x, y, w, h = cv.boundingRect(conts[0])

        cv.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

        cx = round(x + w / 2)

        cy = round(y + h / 2)
```

```
        cv.circle(img, (cx, cy), 20, (0, 0, 255), 2)
        mouseloc = mlocold + ((cx, cy) - mlocold) / damfac
        mouse.position = (round(sx - mouseloc[0] * sx / camx), round(mouseloc[1] * sy / camy))
        mlocold = mouseloc


    cv.imshow("cam", img)
    cv.imshow("mask",mask)
    cv.imshow("mask open",mask_open)
    cv.imshow("mask close", mask_close)
    cv.waitKey(10)
```

8. RESULT AND EVALUATION :

In this paper, we tried to focus on help patients who don't have control of their limbs and also improvise the interaction between the machine and humans. Our motive was to create this technology in the cheapest possible way and also to create it under a standardized operating system.

The proposed system controls the functions of mouse pointer by detecting red, and perform the mouse functions such as left click, dragging, cursor movement, and the file transfer between two systems in a same network. This method detect the red colour objects for the mouse control. The user uses the red color objects on their finger tip for the better performance. When the number of contours are two, then it perform the simple mouse movement action. Otherwise, when the number of contour is one then it perform the left click. This system also supports the simple file transfer between two or more systems in the same network connection. The left side of the computer's screen act as a communication channel between the systems. ie, the file which is to be copy should drag and drop on the left side of the computer's screen. Then the dropped file will be copied to the destination or the reciever system.

This system is mainly aimed to reduce the use of hardware components attatched with the computer. Although the application can be run in a ordinary computer having a web camera. but ideally it requires having at least 2MP frontal camera with at least Pentium processor and at least 256 MB RAM. We can see these all operations on the following images:

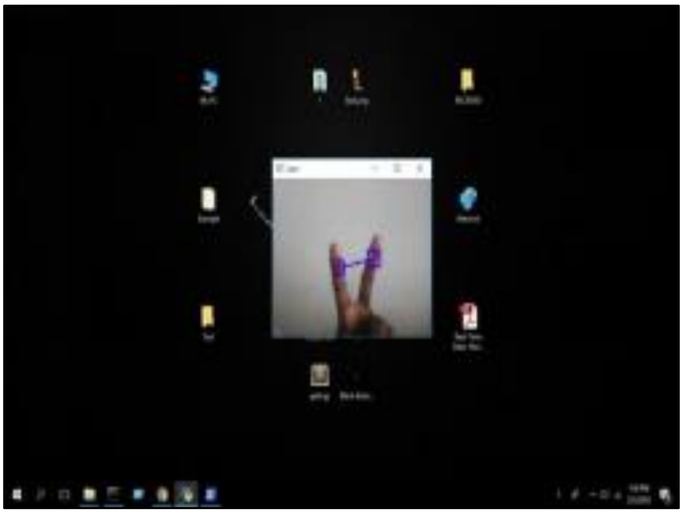


Fig 1 : Moving the mouse pointer

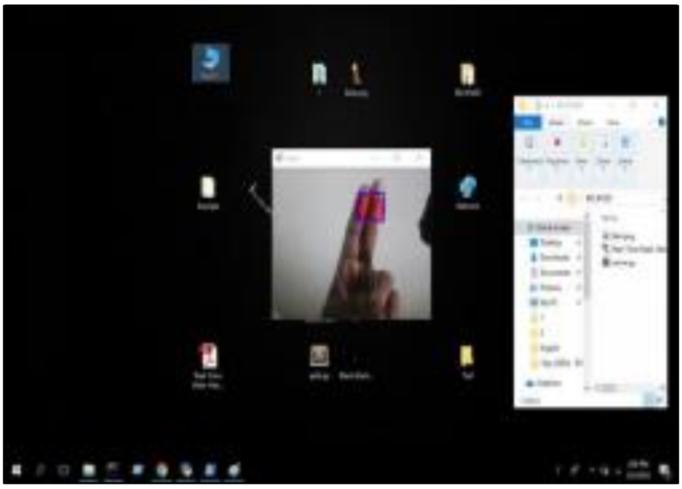


Fig 2 : Performing mouse click

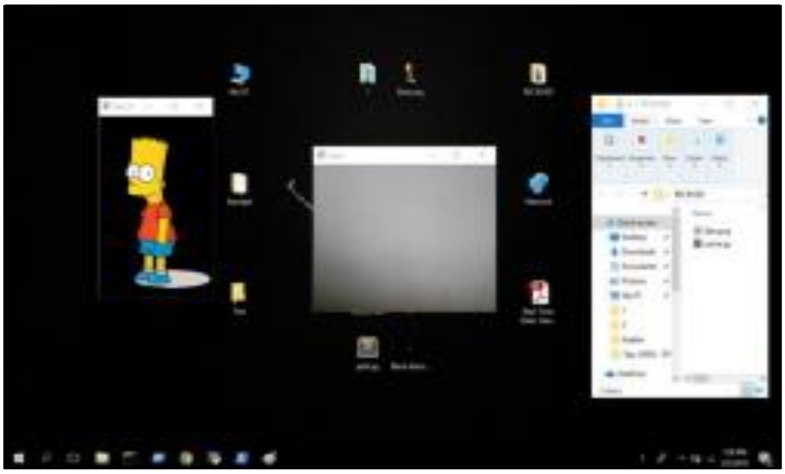


Fig 3 : Performing image file transfer

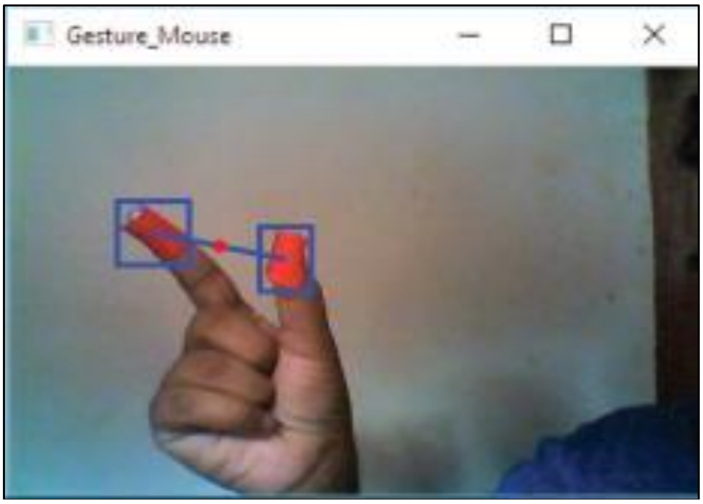


Fig. 4. Mouse Movement (Open Gesture)

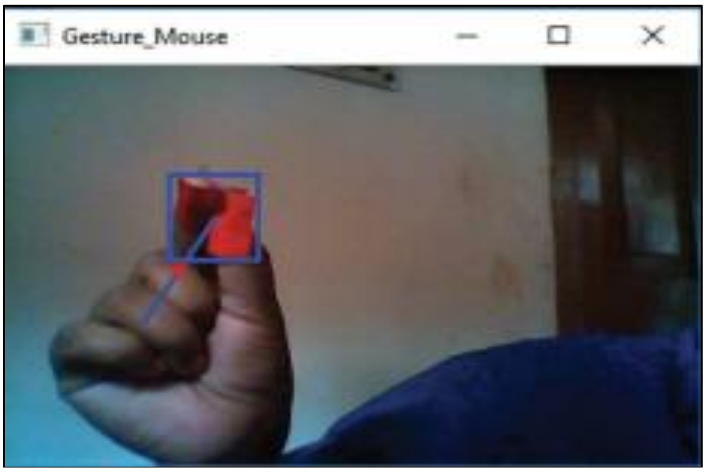


Fig. 5. Left Button Click (Close Gesture)

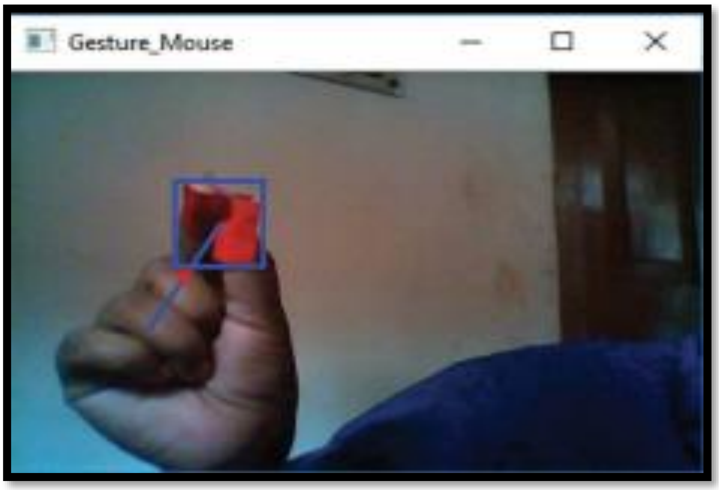


Fig. 6 Seconds Perform Double Click (Close Gesture)

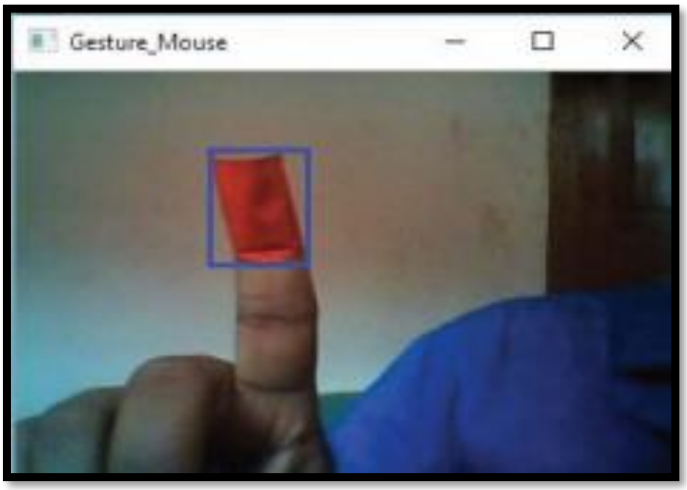
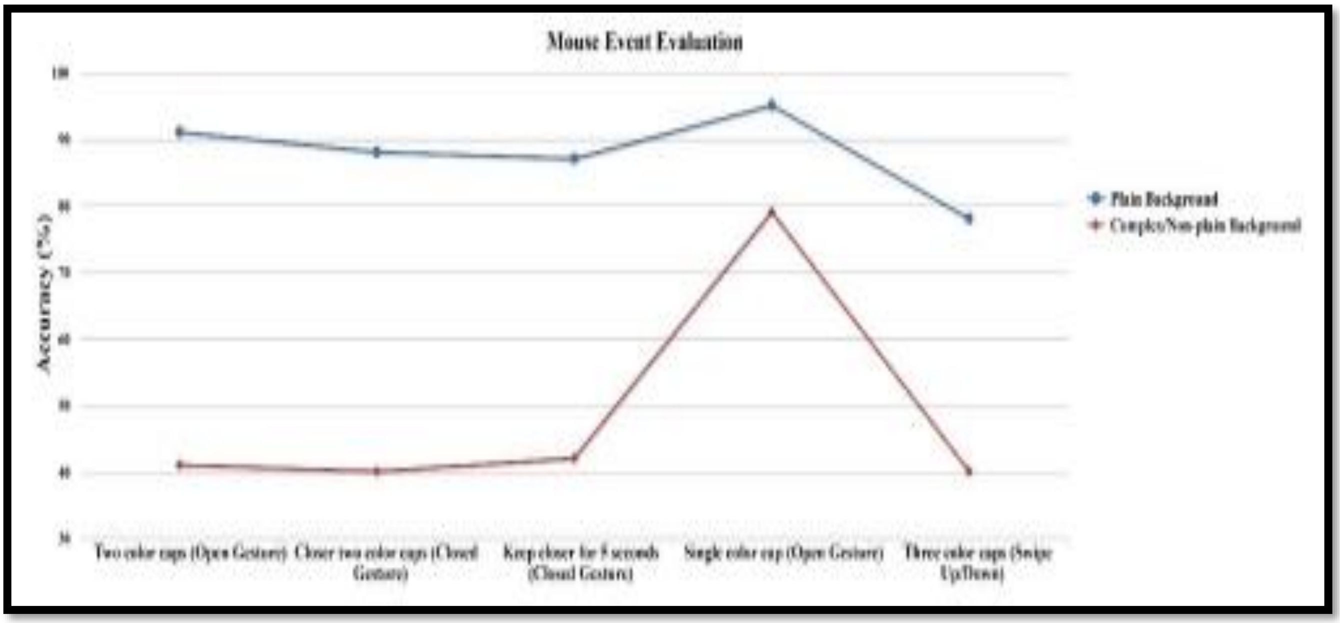


Fig. 7. Right Button Click (Open Gesture)

Mouse Event Accuracy(%) Evaluation



9.Libraries In Use

The external libraries that we will be using:

- openCV
- numpy
- wx
- pynput

10.Features of OpenCV Library

Using OpenCV library, you can –

- Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.

Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it

11. CONCLUSIONS :

Gesture recognition gives the best interaction between human and machine. Gesture recognition is also important for developing alternative human computer interaction modalities. It enables human to interface with machine in a more natural way. Gesture recognition can be used for many applications like sign language recognition for deaf and dumb people, robot control etc.

This technology has wide applications in the fields of augmented reality, computer graphics, computer gaming, prosthetics, and biomedical instrumentation. Digital Canvas is an extension of our system which is gaining popularity among artists, by which the artist could create 2D or 3D images using the Virtual Mouse technology using the hand as brush and a Virtual Reality kit or a monitor as display set. This technology can be used to help patients who don't have control of their limbs. In case of computer graphics and gaming this technology has been applied in modern gaming consoles to create interactive games where a person's motions are tracked and interpreted as commands.

The major extension to this work can be done to make system able to work at much complex background and compatible with different light conditions. It can be made as an effective user interface and which can include all mouse functionalities. And also, it would be ideal to research into advanced mathematical materials for image processing and investigate on different hardware solutions that would result in more accurate hand detections. Not only did this project show the different gesture operations that could be done by the users but it also demonstrated the potential in simplifying user interactions with personal computers and hardware systems.

12. REFERENCES:

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

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

[3] Q. Y. Zhang, F. Chen and X. W. Liu, “Hand Gesture Detection and Segmentation Based on Difference Back ground Image with Complex Background,” Proceedings of the 2008 International Conference on Embedded Soft ware and Systems, Sichuan, 29-31 July 2008, pp. 338- 343.

[4] P.S. Neethu,”Real Time Static & Dynamic Hand Gesture Recognition,” International Journal of Scientific & Engineering Research, Vol. 4, Issue3, March 2013.