ABSTRACT

Talking about gestures makes us return to the historical beginning of human communication, because, in fact, there is no language completely free of gestures. People cannot communicate without gestures. Any action or movement without gestures is free of real feelings and cannot express the thoughts. The purpose from any hand gestures recognition system is to recognizes the hand gesture and used it to transfer a certain meaning or for computer control or and device. Our paper introduced a low-cost system to recognize the hand gesture in real-time. This project describes efforts towards new approach for Human Computer Interaction (HCI) and to control the mouse cursor movement and click events of the mouse using hand gestures. The applications of real time hand gesture recognition in the real world are numerous, due to the fact that it can be used almost anywhere where we interact with computers. Hand gesture depend upon camera-based color detection technique. This method mainly focuses on the use of a Web Camera to develop a virtual human computer interaction device in a cost-effective manner. Current methods involve changing mouse parts such as adding more buttons or changing the position of the tracking ball. Instead, our method is to use a camera and computer vision technology, such as image segmentation and gesture recognition, to control mouse & keyboard tasks (left and right clicking, double-clicking, scrolling and functions of keyboard) and we showed that how it can perform everything as current mouse & keyboard devices can.

TABLE OF CONTENTS

Contents **Page No.**

Chapter 1

Introduction

Chapter2

System Analysis

Chapter 3

System Design

Chapter 4

System Development

Chapter 5

User's Guide

Chapter 6

Conclusion

Chapter 1

INTRODUCTION

1.1 Project Introduction:

Historically, the Electronic Visualization Lab was the first to create a data glove called Sayre Glove this was 1977. Thirty-five years later, the researchers adopted the camera to interact with the computer. In fact, the camera compared to the data glove is considered more direct and natural to achieve Human Computer Interaction (HCI).

Recently, the interactive by gesture has become widely used and, in the future, may replace the mouse and/or keyboard by vision-based devices. The main feature of using hand gestures is to interact with the computer as an input unit. The gesture is defined as a form of non-verbal communication or non- vocal communication where the body's movement can convey certain messages. Gestures originate from different parts of the human body, but the most common ones emerge from the hand or face. Human Computer Interaction (HCI) today greatly emphasizes on developing more spontaneous and natural interfaces.

The Graphical User Interface (GUI) on Personal Computers (PCs) is quite developed, well defined an efficient interface for a user to interact with the computer and access the various applications effortlessly with the help of mouse, track pad, etc. In the present-day scenario, most of the mobile phones are using touch screen technology to interact with the user. But this technology is still not cheap to be used in desktops and laptops.

Our objective was to create a virtual mouse & keyboard system using Web camera to interact with the computer in a more user-friendly manner that can be alternative approach for the touch screen or keyboard. Touch screen are also a good control interface and nowadays it is used globally in many applications. But by applying vision technology and controlling the mouse & keyboard by natural hand gestures, reduce the workspace required.

On developing this approach that uses a video device to control the mouse & keyboard system. This system can control all mouse & keyboard tasks, such as clicking (right and left), cursor movement, double clicking, scrolling and function of keyboard

1.1.1 Problem Statement:

Almost people use a desktop or laptop these days. One of serious problem is that we are stuck to the keyboard and mouse, which will cause a serious health problem on a long run. Moreover, in VR/AR age, we cannot use mouse & keyboard. Our purpose is to replace a mouse & keyboard with hand gestures. We have devised a virtual mouse & keyboard with subtle hand gestures and made Open CV commands recognizes our gestures so that we can control our computer remotely. Amazon echo has ear now. It will have eye in future. We need to make a standard gesture for people to adopt easily like a standard mouse & keyboard.

1.1.2 Scope of the project:

For the future work, swipe keypads which detect the gestures in air view can also be implemented. This would improve the results when the typing speed is quick. The same keypad can be made multi-lingual by just changing the keys on the keypad. Also, this technique can be further improvised to be used on a smart TV.

There are generally two approaches for hand gesture recognition, which are hardware based, where the user must wear a device, and the other is vision-based, which uses image processing techniques with inputs from a camera. The proposed system is vision based, which uses image processing techniques and inputs from a computer webcam. The scope of the project would therefore be to design a vision-based Cursor & Mouse Control system, which can perform the mouse 7 keyboard functions previously stated.

1.1.3 Objectives of the project:

The objectives of this project are:

- Research existing methods and accessories for cursor control and the suitability of visually based methods for cursor control.
- Investigate tracking algorithms utilized for cursor control.
- Develop and implement a computer application that utilizes alternate methods for cursor control.
- Compare the accuracy and precision of the application with alternative accessories

1.1.4 Motivation:

Most laptops today are equipped with webcams, which have recently been used in security applications utilizing face recognition. In order to harness the full potential of a webcam, it can be used for vision-based Cursor & keyboard Control, which would effectively eliminate the need for a computer mouse & keyboard. The usefulness of a webcam can also be greatly extended to other HCI application such as a sign language database or motion controller.

Over the past decades there have been significant advancements in HCI technologies for gaming purposes, such as the Microsoft Kinect and Nintendo Wii. These gaming technologies provide a more natural and interactive means of playing videogames. According to Benedetti, motion controls is the future of gaming and it have tremendously boosted the sales of video 12 games, such as the Nintendo Wii which sold over 50 million consoles within a year of its release.

1.2 Conclusion and Future Prospectus:

The system that provided used only bare hand and webcam of Laptop, so it is very flexible for the user. The system does not need a database, but it is directly distinguishing the gesture, and this achieves the speed of the system. The results of system shown that the rate of recognition is 96.6% and this result is considered very good compared with other research papers. In future and to enhance the system can use both hands instead of use only right hand and that will increase the number of gestures. The results section showed that the best rate of recognition is when the background is clear and the light is medium, so these limits must be addressed in the future in order to increase the accuracy of the system. The vision-based cursor & keyboard control using hand gesture system was developed in the Python language, using the OpenCV library. The system was able to control the movement of a cursor & keyboard by tracking the user's hand. Cursor & keyboard functions were performed by using different hand gestures. The system has the potential of being a viable replacement for the computer mouse & keyboard. However due to the constraints encountered; it cannot completely replace the computer mouse & keyboard. The major constraint of the system is that it must be operated in a well-lit room. This is the main reason why the system cannot completely replace the computer mouse & keyboard, since it is very common for computers to be used in outdoor environments with poor lighting condition. The accuracy of the hand gesture recognition could have been improved, if the Template Matching hand gesture recognition method was used with a machine learning classifier. This would have taken a lot longer to implement, but the accuracy of the gesture recognition could have been improved. All the objectives of the project were completed with satisfactory results. The "*Gesture Controlled Windows Mouse & keyboard*" has great prospects as this method can be used for keyboard functionality control through hand gestures using Machine learning and Deep Learning tools.

Chapter 02

System Analysis

2.1 Feasibility Study:

Many researchers in the human computer interaction and robotics fields have tried to control mouse movement & use of keyboard using video devices. However, all of them used different methods to make different events. One approach used fingertip tracking to control the motion of the mouse. A click of the mouse button was implemented by defining a screen such that a click occurred when a user's hand passed over the region. Another approach was developed that used only the fingertips to control the mouse cursor and click. His clicking method was based on image density and required the user to hold the mouse cursor on the desired spot for a short period of time. Another method used the motion of the thumb (from a 'thumbs-up' position to a fist) to mark a clicking event thumb. Movement of the hand while making a special hand sign moved the mouse pointer. In some research paper, someone has represented some of the innovative methods of the finger tracking used to interact with a computer system using computer vision.

They have divided the approaches used in Human Computer Interaction (HCI) in two categories: 1. HCI without using interface and 2. HCI using interface. Moreover, they have mentioned some useful applications using finger tracking through computer vision. This method uses gesture based interactive experiment using finger movements to stimulate mouse & keyboard operations. As compared with the traditional segmentation method this method has two benefits one is that it uses colored tape, and another is that it requires no special object model with relative high performance. These two benefits make the system applicable to the augmented reality systems or other real-time systems.

Since the system is based on image capture through a webcam, it is dependent on illumination to a certain extent. Furthermore, the presence of other colored objects in the background might cause the system to give an erroneous response. Although by configuring the threshold values and other parameters of the system this problem can be reduced but still that the operating background be light, and no bright colored objects be present. The system might run slower on certain computers with low computational capabilities because it involves a lot of complex calculations in a very small amount of time. However, a standard pc or laptop has the required computational power for optimum performance of the system. Another fact is that if the resolution of the camera is too high then the system might run slow. However, this problem can be solved by reducing the resolution of the image by making changes in the system. Our project is also inspired by this approach and

used Web camera to detect color tapes for cursor movement. The clicking actions were performed by calculating the distance between two colored tapes in the fingers.

Application:

The application is based on Python 3.7 that uses OpenCV for Gesture Control and uses PyAutoGUI library for performing mouse functions. In order to run the application, user should have above mentioned dependencies installed on the computer. The rest of the details about the application is given on "*User's Guide*" and "*Methodology section*".

2.2 Existing System:

Data Gathering:

The mouse & keyboard is the primary input device used to interact with objects in Windows. Mouse functionality can also encompass other pointing devices, such as trackballs, touchpads and pointing sticks built into notebook computers, pens used with Windows Tablet and Touch Technology, and, on computers with touchscreens, even a user's finger. Mouse devices often have a primary button (usually the left button), a secondary button (usually the right), and a mouse wheel between the two. By positioning the pointer and clicking the primary and secondary buttons on the mouse, users can select objects and perform actions on them. For most interactions, pressing a mouse button while the cursor is over a target indicates the selected target, and releasing the button performs any action associated with the target.

There are different types of mouse attached with PC's to control cursor movements. Some of these mouse are Trackball mouse, Optical mouse and wireless optical mouse also external and attached keyboard.

2.3 Existing System:

Data Analysis:

According to existing system data analysis, most of the people use attached mouse & keyboard as an input device or use mouse pad & keyboard already placed on laptop. It is still an effective way of controlling the cursor control and mouse functionalities. However, in current world of technology advancements where things are replaced a day after another with different inventions, efforts have been made to control cursor controlling using hand gestures. We have also made an effort to control cursor using gestures.

2.3.1 Requirement engineering:

User Interfaces

Due to the nature of this application, there are no GUIs to interact with the user. The options which the user has are to turn the program on and off.

Software Interfaces

This application uses the OpenCV library on the Microsoft Visual C++ 2010 platform. At this time, the application can only be used on the Windows operating system. This application does not need to communicate with any other programs on the computer.

Functional Requirements

The first function of this application is to control the movement of the cursor. The second function is using the functionalities of keyboard. The application does this by firstly using skin detection to segment all the skin color pixels in the image. Edge detection will then be performed to find the hand contour in the image. Using shape analysis, the tip of the index finger was found. This fingertip would be tracked and used to control the movement of the cursor & function of keyboard. The next function that the application of the program is to recognize the hand gesture the user makes. Using the contour of the hand, the shape analysis method known as convexity defects is used to recognize the hand gesture. A particular hand gesture is mapped to a mouse function such as left and right clicks. The closed fist gesture will stop the program from executing. If the user makes a hand gesture that the program does not recognize, the program does nothing and will continue to track the user's hand.

Non-Functional Requirements

The main function of the application is to control the cursor & keyboard functions of the computer. The main requirement would therefore be to ensure that the application can control the movement of the cursor & keyboard and perform the primary cursor & keyboard functions such as right click, left click, scroll up, scroll down, screenshot, select all, print text, copy text, close window and so on.

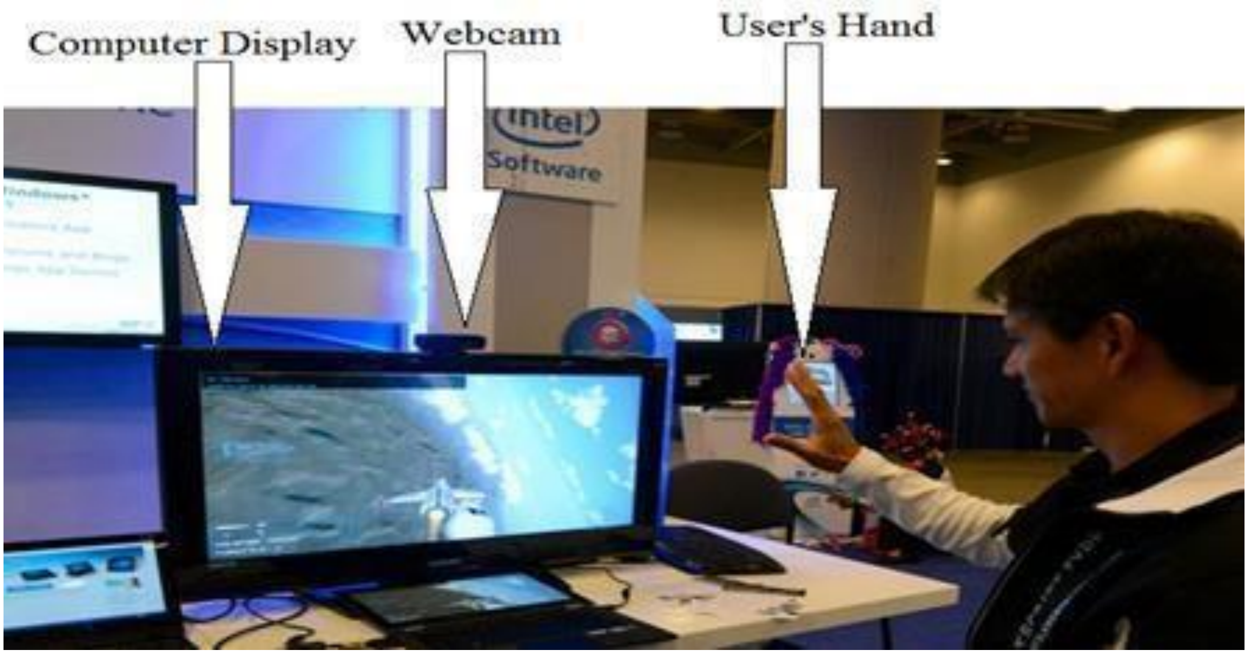
Chapter 3

System Design

3.1 Overall Description

3.1.1 Product Description:

In the overview of the system, the suggested environmental setup of the system is described as well as the architecture of the system. The suggested environmental setup for the system is shown in the figure below.



The user’s hand should be at least 24cm from the webcam and the room should be well lit. The webcam must not be moving and if it is shifted the program should be restarted. In the figure above, the user is only using one hand to operate the program and the other hand must not be in front of the webcam.

3.1.2 Product Functionality:

- Control the movement of the computer cursor
- Control mouse functions such as left click, right click, double click scroll up and scroll down
- Control the functionalities of keyboard such as screen shot, window close, volume up/down, paragraph, select all and so on.

3.1.3 Users and Characteristics:

The two main categories of users for this product are typical users and expert users. The typical user would be anyone who users the application for their personal use such as navigating the computer cursor. This product was originally intended of this type of user and the product would be most frequently used by them. The expert user would incorporate the application into their design or product. Robotic engineers would be classified as an expert user, since they would incorporate this application into their robot control application. Computer game developers would also be considered as an expert user since they can also incorporate this application into their game design.

3.1.4 Operating Environment:

This application uses color information to detect skin color from the webcam image. The environment in which this application is being used must be well lit. The user's hand should be free from any paint or markings that would obscure skin detection. The light source should be white and not be colored, as this would cause the appearance of skin color to change. There should not be any moving persons in the background the application is being used. The lighting condition of the room should be constant.

3.1.5 Design and Implementation Constraints:

The major limitation of this software is that it must be used in a well-lit room. If the lighting condition is too poor, then the accuracy of the program sharply drops to a point where it is unusable. The application uses shape analysis to recognize the hand gesture, thus the user's hand must be at least 24cm from the webcam. Another constraint was that in the background subtraction, the background is learnt and if the camera is moved the background must be learnt again by restart the application. The users face should remain relatively stationary compared to their hand.

3.1.6 User Documentation:

At this moment there aren’t any user-manuals or online support for this application. In the future, a website will be created to allow users to share their problems and report any bugs in the program. It will also allow users to download updates for the software.

3.1.7 Assumptions and Dependencies:

It is assumed that the user has a computer with a working webcam. The program will not work without a working webcam. The application will automatically attempt to turn on the webcam and if this fails the user will be prompted to manually turn on the computers webcam. It is also assumed that the resolution of the webcam is at least 480 pixels.

3.2 Methodology:

Following are the steps in our approach:

a. Capturing real time video using Web-Camera.

b. Processing the individual image frame.

c. Flipping of each image frame.

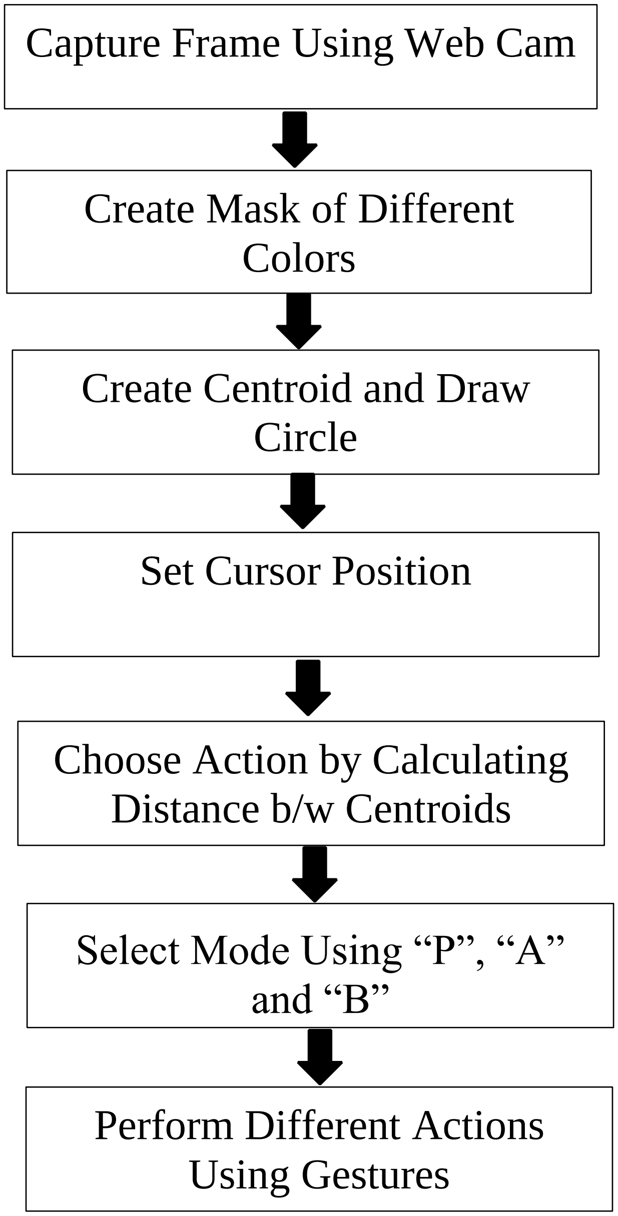
d. Color detection and extraction of the different colors (RGB) from flipped image.

e. Conversion of the detected image into a HSV values.

f. Finding the region of the image and calculating its centroid.

g. Tracking the mouse pointer using the coordinates obtained from the centroid.

Simulating the left click and the right click events of the mouse & keyboard by assigning different color pointers.


Capture Frame Using Web Cam
Create Mask of Different Colors
Create Centroid and Draw Circle
Set Cursor Position
Choose Action by Calculating Distance b/w Centroids
Select Mode Using “P”, “A” and “B”
Perform Different Actions Using Gestures

Basic Block Diagram

3.2.1 Capturing the real time video:

For the system to work we need a sensor to detect the hand movements of the user. The webcam of the computer is used as a sensor. The webcam captures the real time video at a fixed frame rate and resolution which is determined by the hardware of the camera. The frame rate and resolution can be changed in the system if required.

- Computer Webcam is used to capture the Real Time Video
- Video is divided into Image frames base on the FPS (Frames per second) of the camera
- Processing of individual Frames

3.2.2 Flipping of Images:

When the camera captures an image, it is inverted. This means that if we move the color pointer towards the left, the image of the pointer moves towards the right and vice-versa. It’s similar to an image obtained when we stand in front of a mirror (Left is detected as right, and right is detected as left). To avoid this problem, we need to vertically flip the image. The image captured is an RGB image and flipping actions cannot be directly performed on it. So, the individual color channels of the image are separated and then they are flipped individually. After flipping the red, blue and green colored channels individually, they are concatenated, and a flipped RGB image is obtained.

3.2.3 Conversion of Flipped Image into HSV values:

As compared to a colored image, computational complexity is reduced in HSV image. Thus, the flipped image is converted into a gray scale image. All the necessary operations were performed after converting the image into gray scale.

3.2.4 Color Detection:

This is the most important step in the whole process. The red, green and blue color object is detected by subtracting the flipped color suppressed channel from the flipped Gray-Scale Image. This creates an image which contains the detected object as a patch of grey surrounded by black space.

3.2.5 Finding Centroid of an object and plotting:

For the user to control the mouse pointer it is necessary to determine a point whose coordinates can be sent to the cursor. With these coordinates, the system can control the cursor movement. An inbuilt function in MATLAB is used to find the centroid of the detected region. The output of function is a matrix consisting of the X (horizontal) and Y (vertical) coordinates of the centroid. These coordinates change with time as the object moves across the screen.

3.2.6 Tracking the Mouse pointer:

Once the coordinates have been determined, the mouse driver is accessed, and the coordinates are sent to the cursor. With these coordinates, the cursor places itself in the required position. It is assumed that the object moves continuously, each time a new centroid is determined and for each frame the cursor obtains a new position, thus creating an effect of tracking. So, as the user moves his hands across the field of view of the camera, the mouse moves proportionally across the screen.

3.2.7 Performing Clicking Actions:

The control actions of the mouse are performed by controlling the colored fingers which is detect by centroids. The user has to perform hand gestures in order to create the control actions. Due to the use of color pointers, the computation time required is reduced. Furthermore, the system becomes resistant to background noise and low illumination conditions.

Chapter 4

System Development

4.1 Hardware involved:

- Laptop or PC Minimum 4th Generation
- Webcam separate or built-in in Laptop

4.2 Software involved:

4.2.1 IDE's:

- IDLE'S (Python 3.7 or 3.8)
- JetBrains PyCham Community

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSs), and supports web development with Django as well as Data Science with Anaconda.

Boost Code Quality

Write neat and maintainable code while the IDE helps you keep control of the quality with PEP8 checks, testing assistance, smart refactoring's, and a host of inspections.

Simply All You Need

PyCharm is designed by programmers, for programmers, to provide all the tools you need for productive Python development.

4.2.2 Programming Languages:

Python 3.7 (Programming Language):

- **Python** is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.
- Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.
- Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), objectoriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.
- Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting.
- Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

OpenCV (Python Library)

- OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.
- The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

- Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV.

  OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

- It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

NumPy (Python Library)

- NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.
- It is the fundamental package for scientific computing with Python. It contains various features including these important ones:
- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities
- Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

PyAutoGUI (Python Library)

PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be as simple. PyAutoGUI works on Windows, macOS, and Linux, and runs on Python 2 and 3.

PyAutoGUI has several features:

- Moving the mouse and clicking or typing in the windows of other applications.
- Sending keystrokes to applications (for example, to fill out forms).
- Take screenshots, and given an image (for example, of a button or checkbox), find it on the screen.
- Locate an application’s window, and move, resize, maximize, minimize, or close it (Windows only, currently)
- Display message boxes for user interaction while your GUI automation script runs.

Chapter 5

User’s Guide

5.1 User’s Guide:

The user manual is developed in such a way that even an un-educated person may use and operate the Application. Some steps have been defined to operate the Application which are as follows:

- Use Yellow tape on Left hand Index Finger
- Use Red tape on Right hand Index Finger
- Use Blue tape on Right hand Middle Finger

There are three modes in the application:

- Mouse Mode
- Keyboard Mode
- Keyboard 1 Mode

Every mode has different functions to be performed. There are different keys to switch between the modes.

Press “P” to turn ON / OFF Mouse Mode.

Press “A” to turn ON / OFF Keyboard Mode.

Press “B” to turn ON / OFF Keyboard 1 Mode.

Step 1: Installed Required Editor & Libraries

- Installed PyCharm Open Source Community
- Installed Python 3.7 Language
- Installed Required Libraries for implementation

After Completion of Above Process, We Can Run the Application

Step 2: Run the Application

- First of all run the given application

Step 3: Centroid Detection

- By pressing “C” Colors are detected and Black Centroid is shown

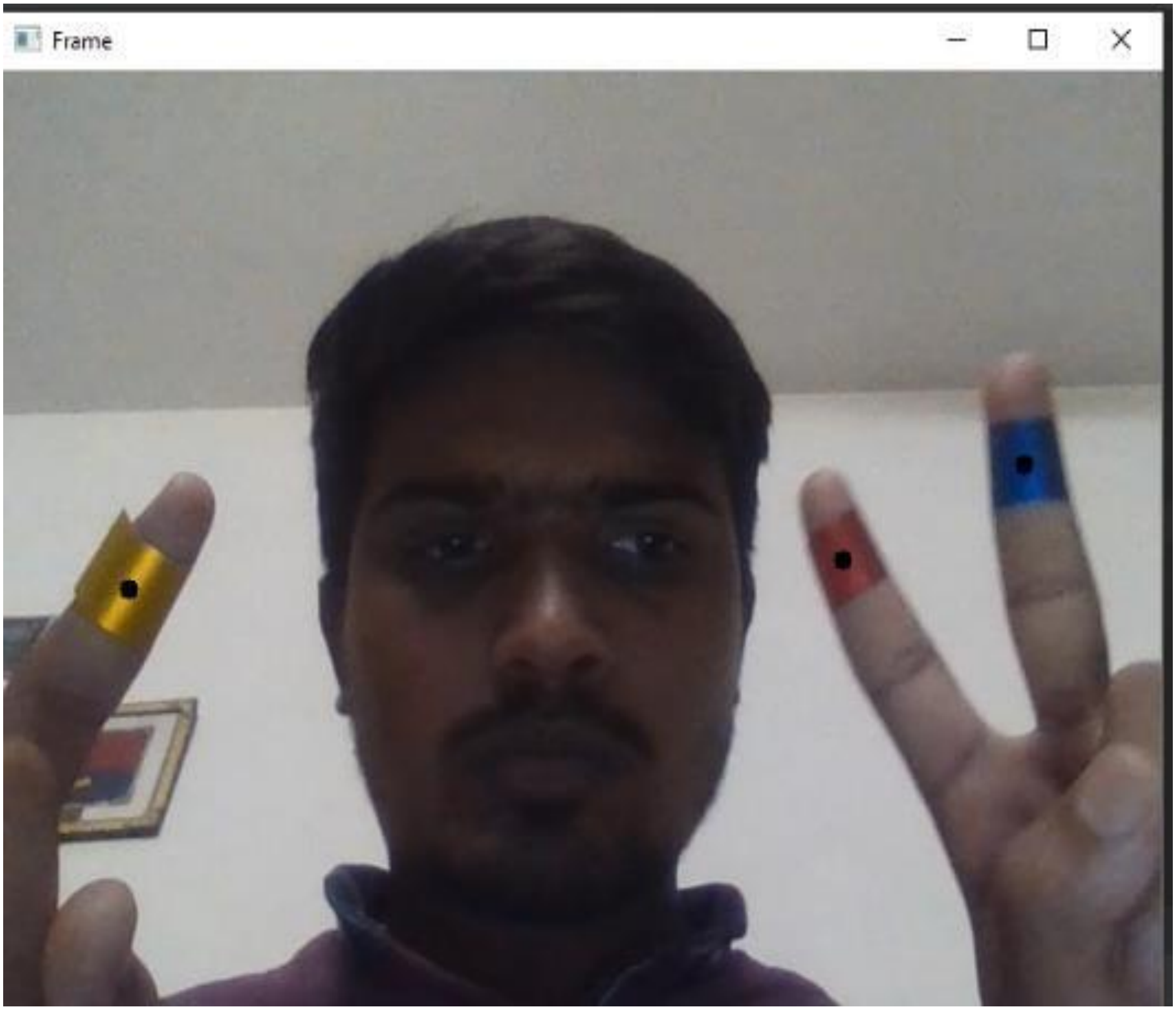
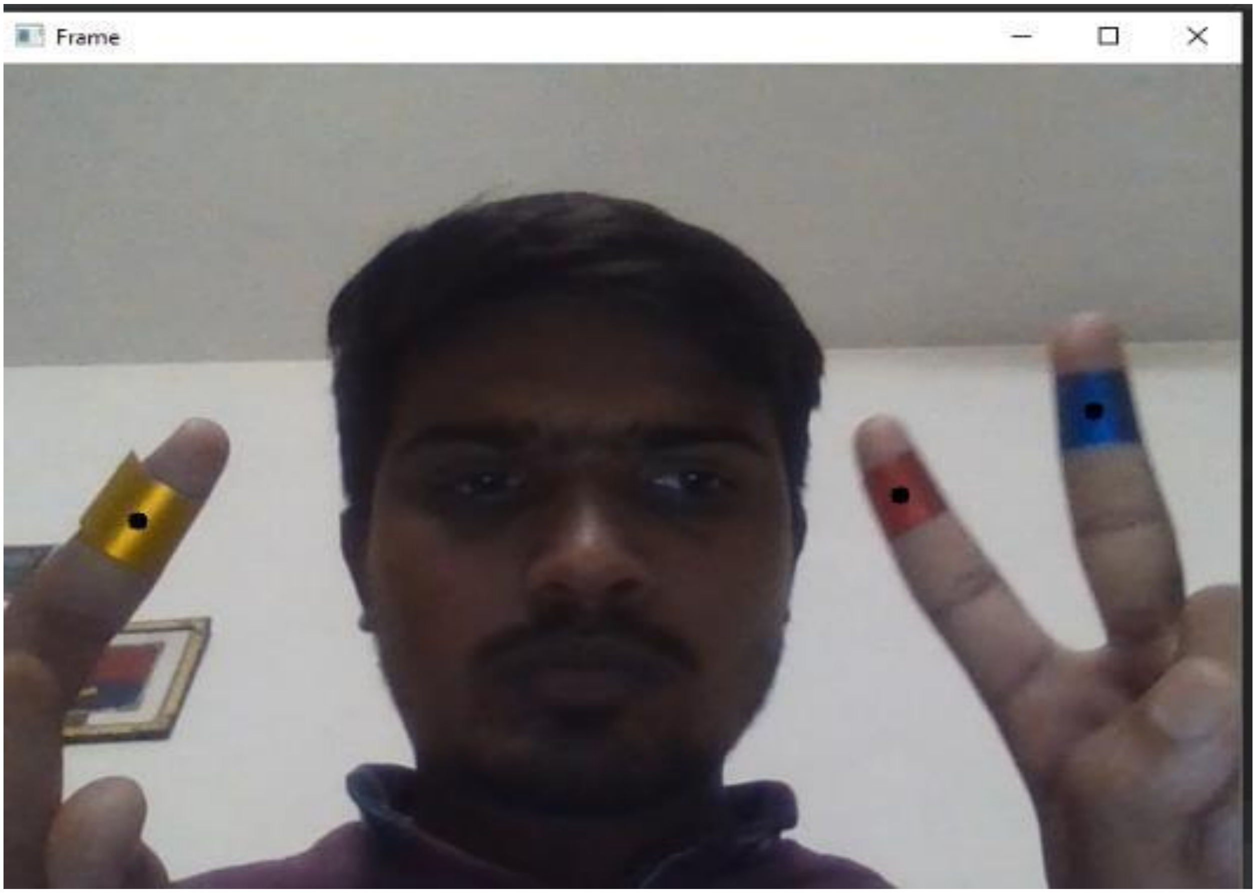


Fig. 5.1.1

Cursor Movement:

- Move the Yellow Centroid or Finger in order to Move the Cursor



4. **Fig. 5.1.2**

Gesture 1:

- Red and Blue finger closed together
- Performs “Left Click” in Mouse Mode, Keyboard Mode and Keyboard 1 Mode

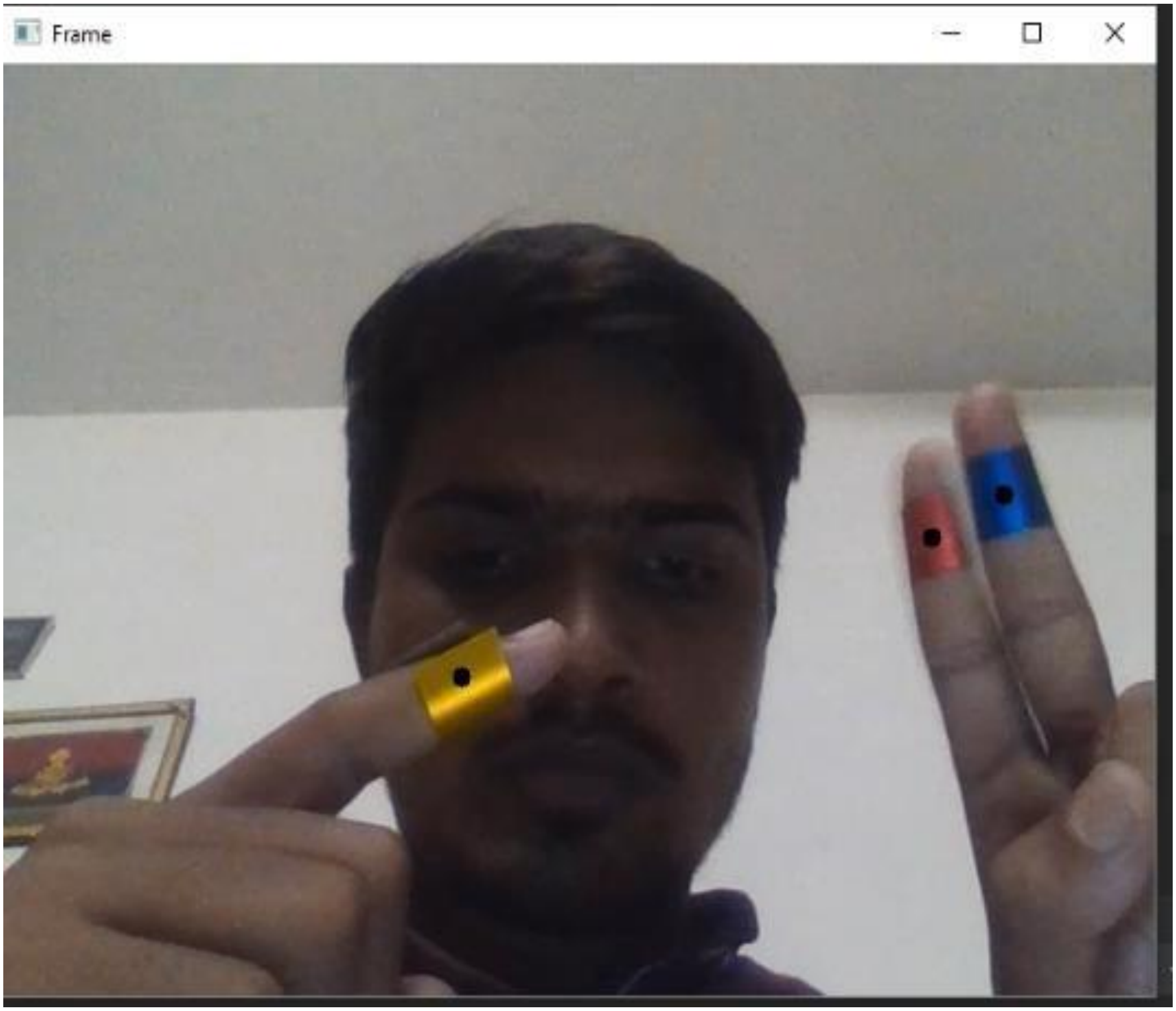


Fig. 5.1.3

Gesture 2:

- Yellow and red centroids closed together
- Performs “Right Click” in Mouse Mode , “Paragraph Writing” in Keyboard Mode and “Screenshot” in Keyboard 1 Mode

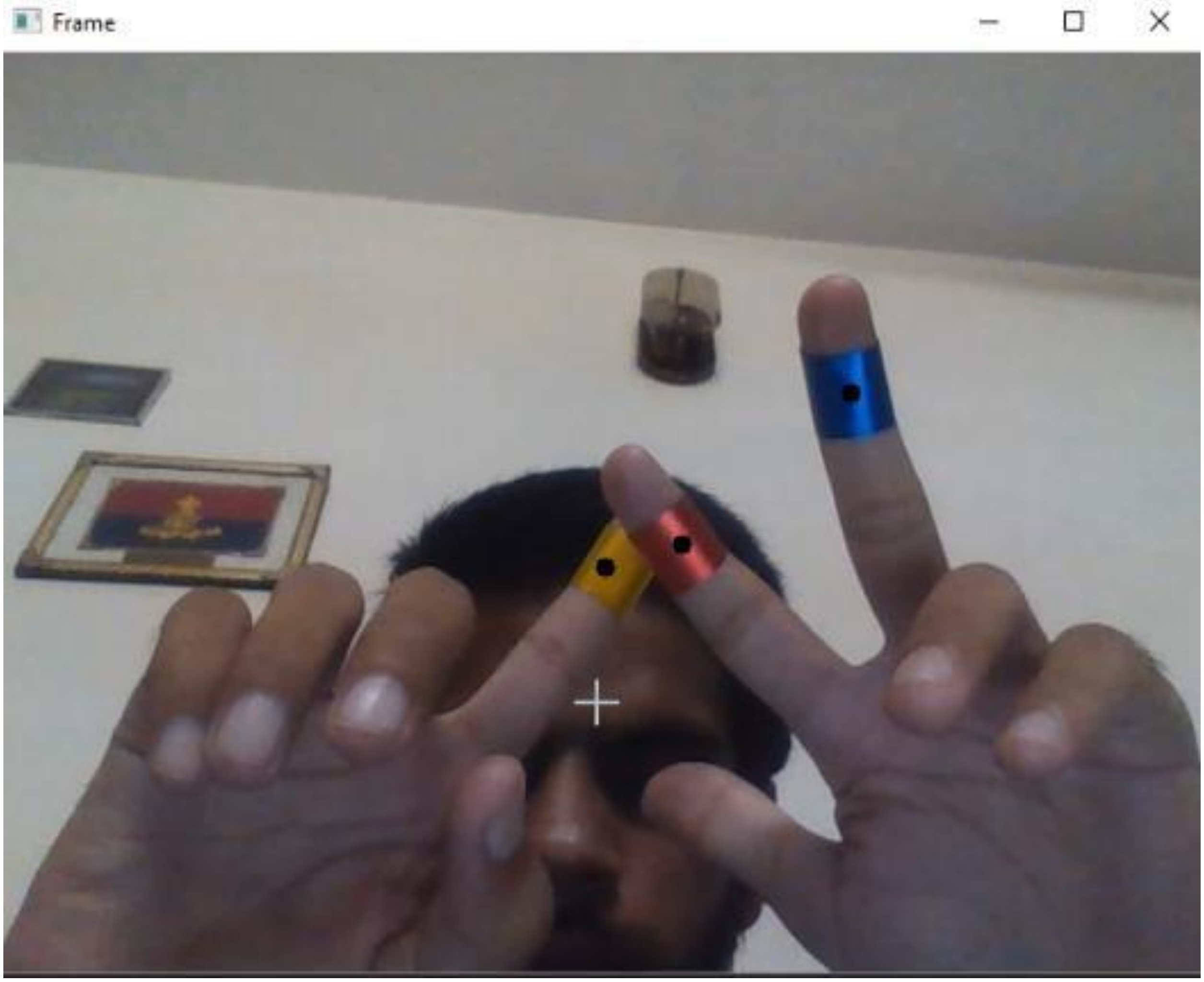


Fig. 5.1.4

Gesture 3:

- Yellow and Blue Centroid closed together
- Performs “Volume Down” in Mouse Mode, “Save” in Keyboard Mode and “Screenshot” in Keyboard 1 Mode

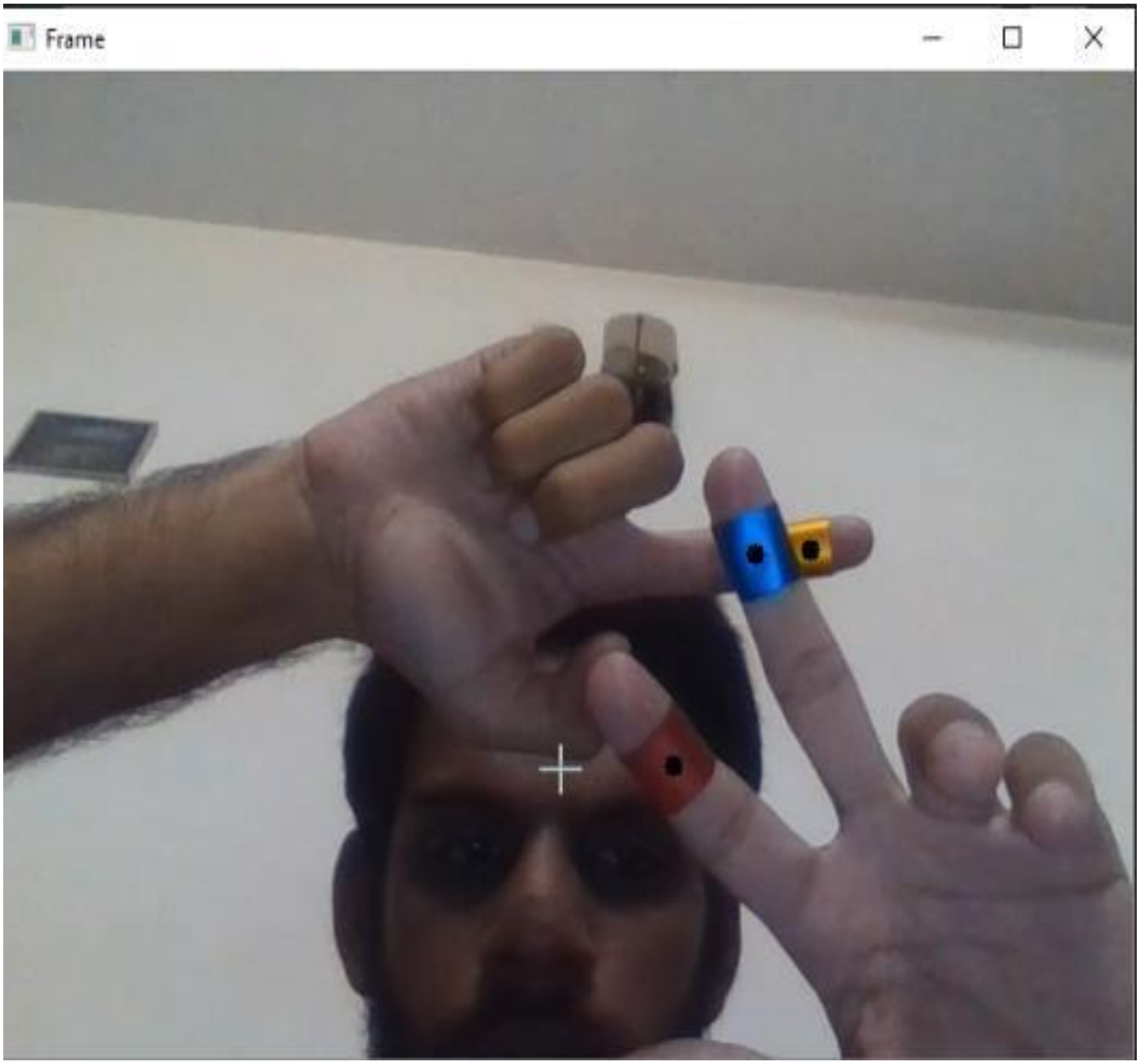


Fig. 5.1.5

Gesture 4:

- Blue and Red Fingers Far Apart along X-axis
- Performs “Volume Up” in Mouse Mode, “Copy” in Keyboard Mode and “Close Window” in Keyboard 1 Mode

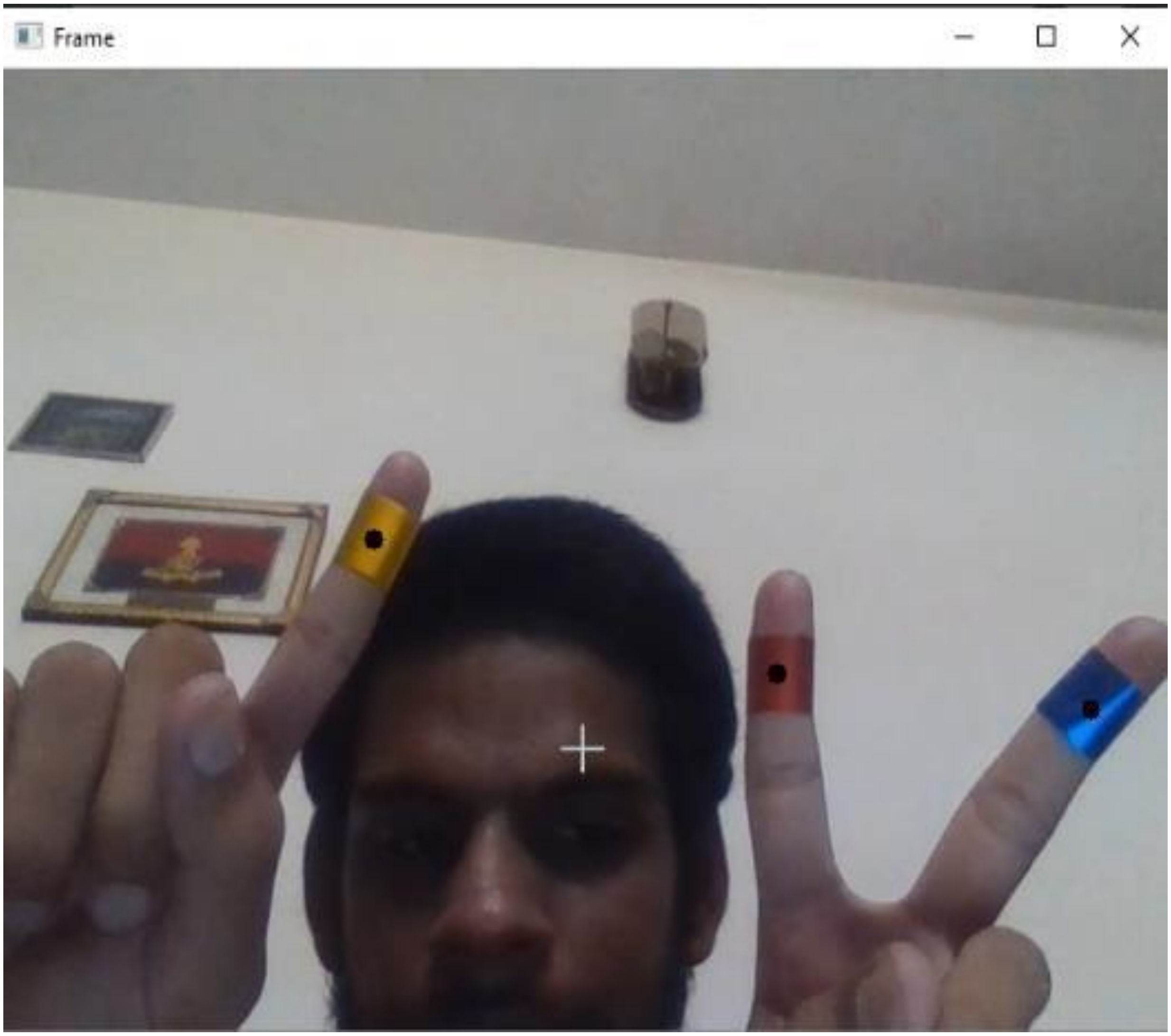


Fig. 5.1.6

Gesture 5:

- Red and Blue Centroids far apart along Y-axis
- Performs “Scroll up” in Mouse Mode, “Select All” in Keyboard Mode and “Maximize” in Keyboard 1 Mode

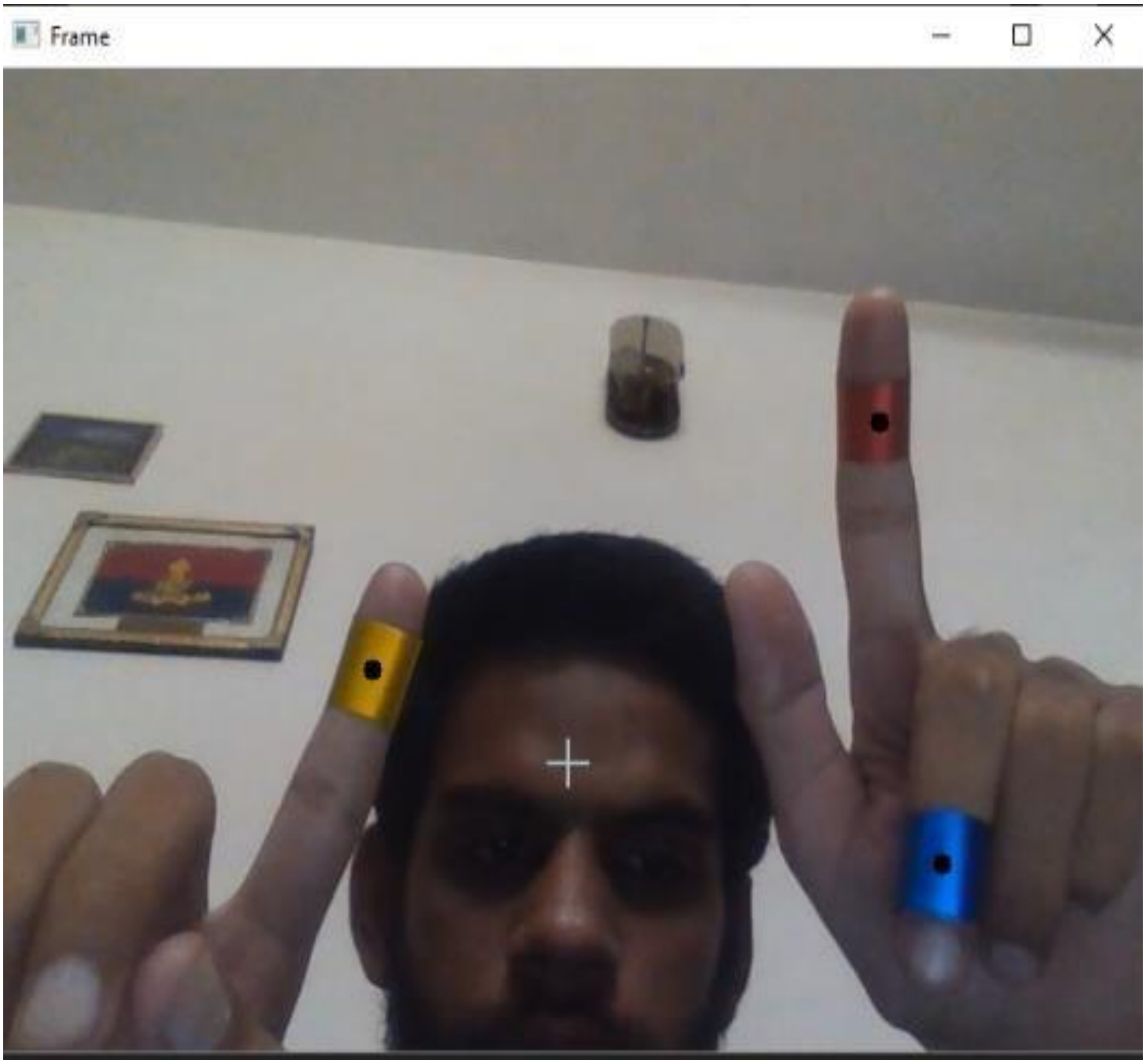
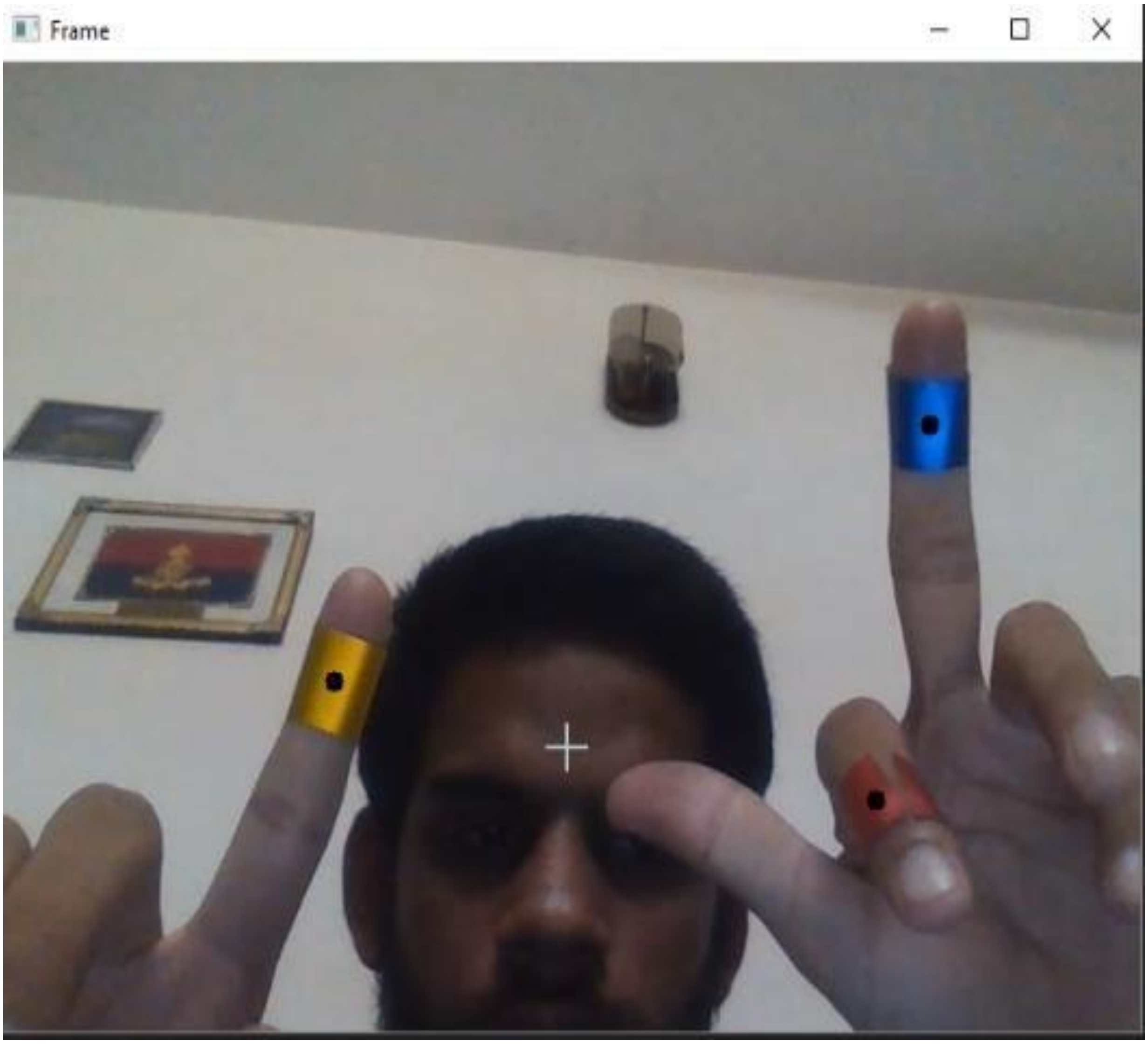


Fig. 5.1.7

Gesture 6

- Blue and Red Centroid Far Apart Along Y-axis
- Performs “Scroll Down” in Mouse Mode, “Paste” in Keyboard Mode and “Minimize” in Keyboard 1 Mode



5. **Fig. 5.1.8**

5.2 Limitations:

- Effect of low intensity and high intensity light. Will not be efficient in a very high intensity or low intensity of light
- Not feasible in the dark
- Uses Webcam. Without camera it will not work
- Uses different colors on fingers to perform actions.
- Background object detection. If background is not clear and has same colors that we have worn on our finger’s then there will be problem in Centroid creation
- Needs well lit room

Chapter 6

Conclusion

6.1 Conclusion:

In this project, we have presented a way of implementing a virtual mouse & keyboard on the screen without any use of external hardware. This drawback which can be further improved by implementing algorithms which allow the web cam in quickly capturing and comparing the consecutive frames will give a better result with less complexity. The mouse & keyboard would recognize a click only if there is a significant movement of the fingertip. If the movement is quick the web cam might not be able to detect it as a click because frames will not be captured quickly and the difference in contour will not be large enough. The Histogram-based and Explicitly threshold skin detection methods were evaluated and based on the results, the Histogram method was deemed as more accurate. The vision-based cursor control using hand gesture system was developed in the Python language, using the OpenCV library. The system was able to control the movement of a cursor 7 functions of keyboard by tracking the user's hand. Cursor functions were performed by using different hand gestures. The system has the potential of being a viable replacement for the computer mouse, however due to the constraints encountered; it cannot completely replace the computer mouse. The major constraint of the system is that it must be operated in a well-lit room. This is the main reason why the system cannot completely replace the computer mouse, since it is very common for computers to be used in outdoor environments with poor lighting condition. The accuracy of the hand gesture recognition could have been improved, if the Template Matching hand gesture recognition method was used with a machine learning classifier. This would have taken a lot longer to implement, but the accuracy of the gesture recognition could have been improved. It was very difficult to control the cursor for precise cursor movements, since the cursor was very unstable. The stability of the cursor control could have been improved if a Kalman filter was incorporated in the design. The Kalman filter also requires a considerable amount of time to implement and due to time constraints, it was not implemented.
All of the objectives of the project were completed with satisfactory results.

6.2 Future Recommendation:

For the future work, swipe keypads which detect the gestures in air view can also be implemented. This would improve the results when the typing speed is quick. The same keypad can be made multi-lingual by just changing the keys on the keypad. Also, this technique can be further improvised to be used on a smart TV. A Kalman filter could have also been used in the Hand Tracking stage. This would have stabilized the tracking of the hand and would have therefore made the application more accurate and easier to use. A hand in mid-air is constantly shaking, which causes the computer cursor to be unstable. The authors also suggested that this instability problem can be solved using a Kalman filter. Another future recommendation for the hand gesture recognition stage would be to use the Template Matching method to distinguish the hand gestures. This method requires the use of a machine learning classifier, which takes a considerably long time to train develop. However, it would have allowed the use of lots more hand gestures which in turn would allow the use of more mouse functions such as zoom in and zoom out. Once the classifier is well trained, the accuracy of the Template Matching method is expected to be better than the method used in the proposed design.

Reference’s

1. C.-Chiung Hsieh, D.-Hua Liou, & D. Lee, , “A real time hand gesture recognition system using motion history image,” Proc. IEEE Int'l Conf. Signal Processing Systems (ICSPS), 2. 10.1109/ICSPS.2010.5555462, 2010.
2. C.-Y. Kao, C.-S. Fahn, "A Human-Machine Interaction Technique: Hand Gesture Recognition Based on Hidden Markov Models with Trajectory of Hand Motion," in Procedia Engineering 15 pp. 3739 – 3743, 2011.
3. P.S. Neethu,”Real Time Static & Dynamic Hand Gesture Recognition,” International Journal of Scientific & Engineering Research, Vol. 4, Issue3, March 2013. 4. A. M. Patil1, S. U. Dudhane1, M. B. Gandhi1,” Cursor Control System Using Hand Gesture Recognition”, International journal of advanced research in computer and communication engineering, vol. 2, issue: 5, May 2013
   5. S. S. Abhilash, L. Thomas, N. Wilson & C. Chaithanya, "VIRTUAL MOUSE USING HAND GESTURE," International Research Journal of Engineering and Technology (IRJET), vol. 05 issue: 4, April 2018.