



# SMART WHITE BOARD

<sup>1</sup>Ashwin Singh(student), <sup>2</sup>Rajinder Singh(student), <sup>3</sup>Sahil Tambe(student),  
<sup>4</sup>Shivam Upadhyay(student), <sup>5</sup>Umesh Mohite (Assistant Professor)

Department of AIML, Universal College of Engineering, Mumbai, India.

## ABSTRACT:

The continuous advancement in technology necessitates the modernization of various sectors. With the evolution of sophisticated devices, it is now possible to virtually control systems through human gestures. When working with painting applications, individuals often encounter challenges in expressing their imagination through traditional methods. The Air Canvas Project addresses this issue by focusing on the development of a motion-to-text converter. This project specifically concentrates on the advancement of a hand tracking system, which serves as a virtual pen for creating drawings and shapes, as well as functioning as an eraser. The implementation utilizes the Open Computer Vision Library (OpenCV) and Mediapipe for efficient hand tracking. In contrast to existing projects that may involve complex processes and time-consuming methodologies for hand-drawn creations, our project leverages new technologies and simplified approaches. The system employs a camera to track hand movements, enabling users to draw or add text effortlessly by waving their hands.

**KEYWORDS:** Air Writing, Mediapipe, OpenCV, Hand Tracking, Gesture-Based Interaction, Computer Vision, Human-Computer Interaction, Machine Learning

## 1. INTRODUCTION

In the past, the task of digital painting was often strenuous and challenging, relying on devices like a mouse or touchpad. Even with the availability of touch screen laptops, their cost can be prohibitive. To alleviate these challenges, hand tracking, specifically finger tracking, has emerged as a valuable tool, functioning as an external device akin to a keyboard and mouse. This technique finds applications in diverse fields, ranging from Virtual Reality to sign language recognition. The Air Canvas project introduces a hands-free digital drawing canvas, utilizing a camera, OpenCV, and Mediapipe to recognize and map hand gestures. In this innovative system, the user's finger serves as the brush or pen for drawing. Python is the chosen language for coding this system, which employs a combination of a camera and Mediapipe for accurate finger position tracking. Computer vision techniques are employed to draw various shapes on the canvas. Additionally, users can add text feature as well as have the option to save their canvas work as an image. The [1] idea was to make the model capable of efficiently recognizing the fingertips of all four fingers. This would allow the user to control the system using the number of fingers he shows.

The [3] virtual paint application presented is based on the frames recorded by the PC's web camera. The frames are captured by the web camera and sent to the system. The method makes use of a web camera to capture each frame till the application is finished. The video frames are transformed from BGR to RGB color to locate the hands in the video frame. The system then determines which finger is up by comparing the tip Id of the corresponding finger found via the MediaPipe to the respective coordinates of the up fingers, and then performs the appropriate function. The user can write anything on the screen if his or her index finger is raised.

## 2. REQUIREMENT SPECIFICATIONS

### 2.1 Hardware Requirements:

This subsection will provide the minimum requirements that must be fulfilled by the hardware components. The hardware requirements are as follows: -

- **Computer:** You will need a computer with sufficient processing power to run computer vision and machine learning algorithms. A modern CPU with multiple cores and a dedicated GPU (optional but recommended) can significantly enhance the performance.
- **Webcam:** A webcam is required to capture the video feed, which is the input for the project. Ensure that your webcam provides good video quality and frame rate.
- **Display:** A computer monitor or screen is needed to display the canvas and interact with the digital environment.
- **Input Device (Optional):** Depending on the project's interactivity, you might need additional input devices such as a keyboard, mouse, or a gesture control device like a Leap Motion controller.

## 2.2 Software Requirements:

This subsection will provide the versions of software applications that must be installed. The software requirements are as follows: -

- **Python:** Ensure you have Python installed on your computer. You can use Python 3.x for this project.
- **OpenCV:** Install the OpenCV library, which is used for computer vision tasks. You can install it via pip.
- **MediaPipe:** Install the MediaPipe library for hand tracking and gesture recognition. You can install it via pip.
- **NumPy:** NumPy is a fundamental package for scientific computing with Python, and it's often used in computer vision projects. Install it using pip.

## 3. METHODOLOGY

This methodology needs a lot of data to be stored and sometimes it leads to wrong prediction due to background difference or the skin color difference. Some of the existing papers followed the process of image processing using threshold values and database data of different images. To overcome all these limitations and drawbacks, we proposed the system using mediapipe. In the proposed system hand tracking is done using mediapipe which first detects hand landmarks and then obtain positions according to it.

Figure 1 describes the flow of system or the process of project in which it works. This is the complete work flow that occurs to execute the system and draw images just by waving hands in efficient manner. [2]

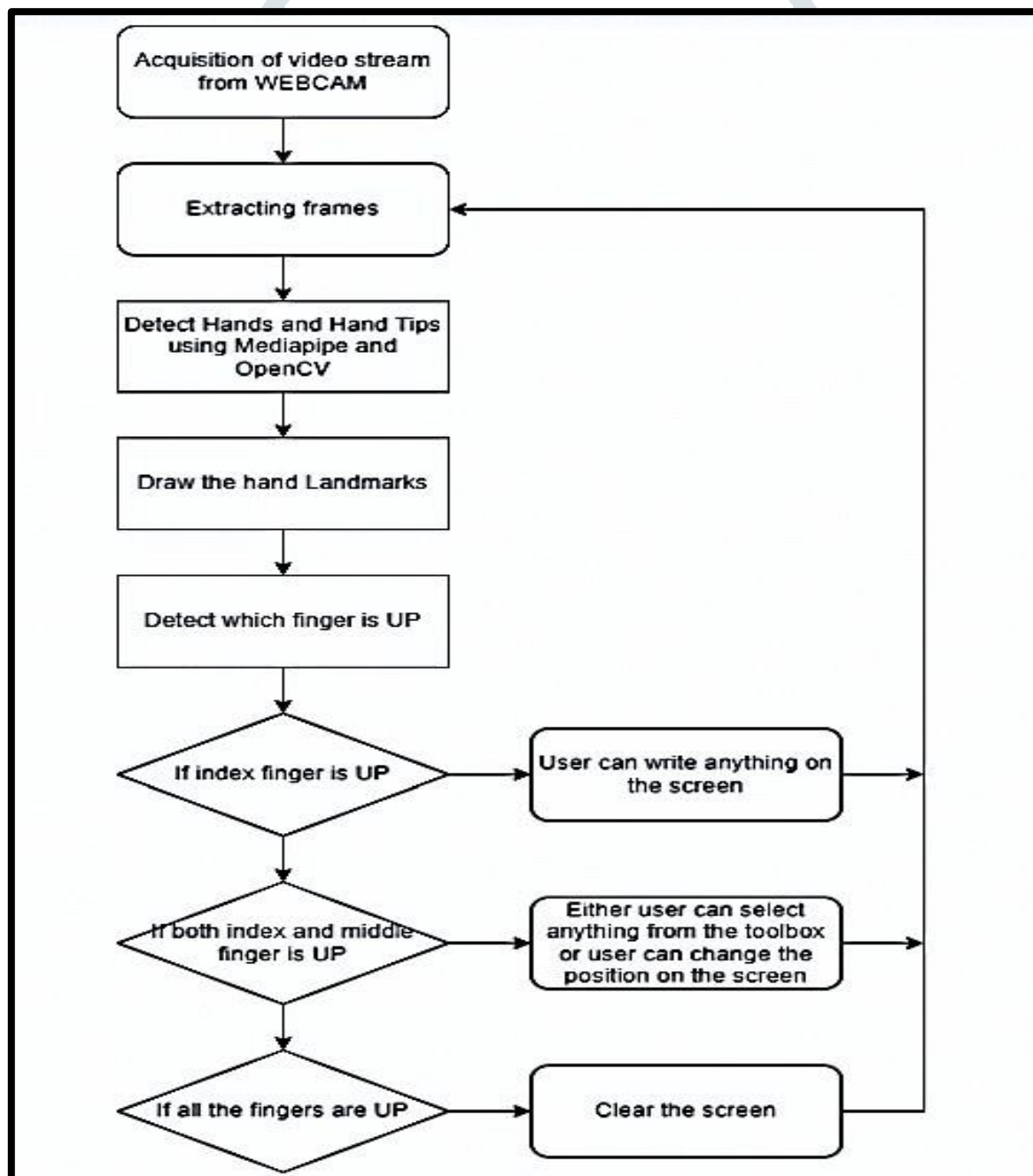


fig.1: flowchart of project

#### 4. USE CASE DIAGRAM

Fig 2 denotes the Use Case Diagram of the proposed system. It shows the user's interaction with the systems. The purpose of a use case diagram in Unified Modelling Language (UML) is to demonstrate the different ways that a user might interact with a system. In this use case diagram, there are two actors involved, the first actor is user, and the second actor is Software Application. It depicts the interactions between the various actors used in this system. All these interactions between actors and system is done in virtual paint environment. The various use cases involved in this system such as draw free style as well as shapes and download content as an image.

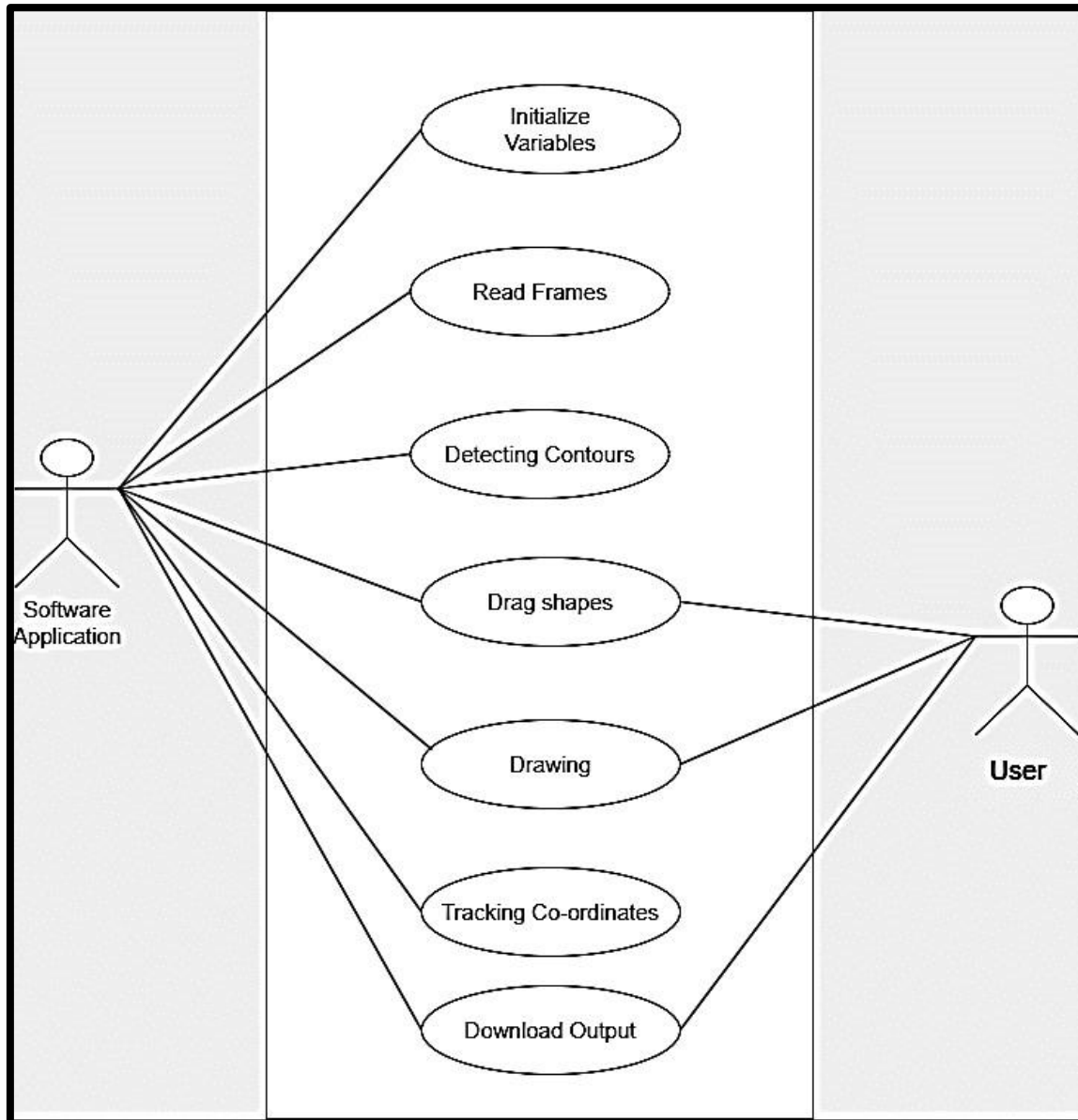


fig.2: use case diagram

#### 5. CASE STUDY: VIRTUAL PAINTING USING MEDIAPIPE, OPENCV

The proposed Air Canvas system utilizing Mediapipe and OpenCV aims to enhance the efficiency of the existing system. The detailed methodologies or stages of the proposed system are explained below [2]:

##### 1. Run or Execute the Code:

The code is executed after ensuring that all the required libraries are installed. This action automatically activates the camera, displaying an OpenCV frame with buttons representing various functions such as shapes, colors, size adjustments, save, clear, erase, etc.

##### 2. Webcam Starts:

The webcam begins recording, capturing each frame of the video. These frames are then sent to the Hand Tracker class to detect and track the positions of fingers. The displayed frame includes buttons for different functionalities, such as selecting shapes and recording video.





fig.3: OpenCV Frame with Buttons.

**3. Detects Hand Landmarks:**

Each received frame is compared with the hand landmarks provided by Mediapipe. The Hand Tracker class utilizes functions like `getPositions()` and `getUpFingers()` to identify the positions of fingers and determine which fingers are open. These functions are integral parts of the Hand Tracker class.



fig.4: Mediapipe Hand Land Marks.

**4. Perform Actions According to Button:**

The system responds to user input based on the selected buttons. The index finger is used to hover over buttons, triggering various actions. Different functionalities are performed on the canvas depending on the selected button, such as drawing shapes, add text or saving.

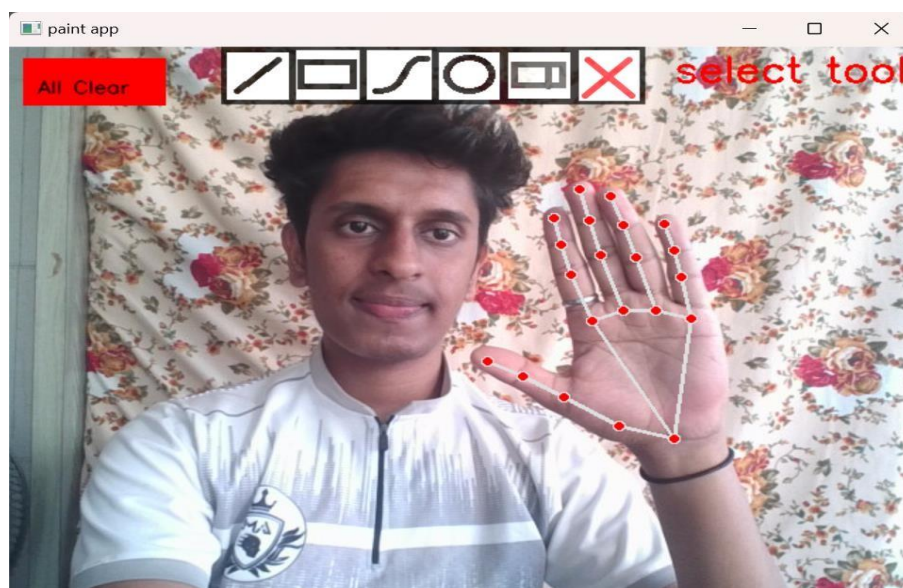


fig.5: Hand Landmarks and buttons.

### 5. Depict on Canvas:

The system enables the depiction of various shapes, including rectangles, circles, ellipses, arrow-headed lines, straight lines, and free-style drawings on the canvas. The distance between the thumb and index finger is used to determine parameters such as diagonal length for rectangles, radius for circles, and length for lines. The canvas also supports clearing or erasing content and saving the created work.

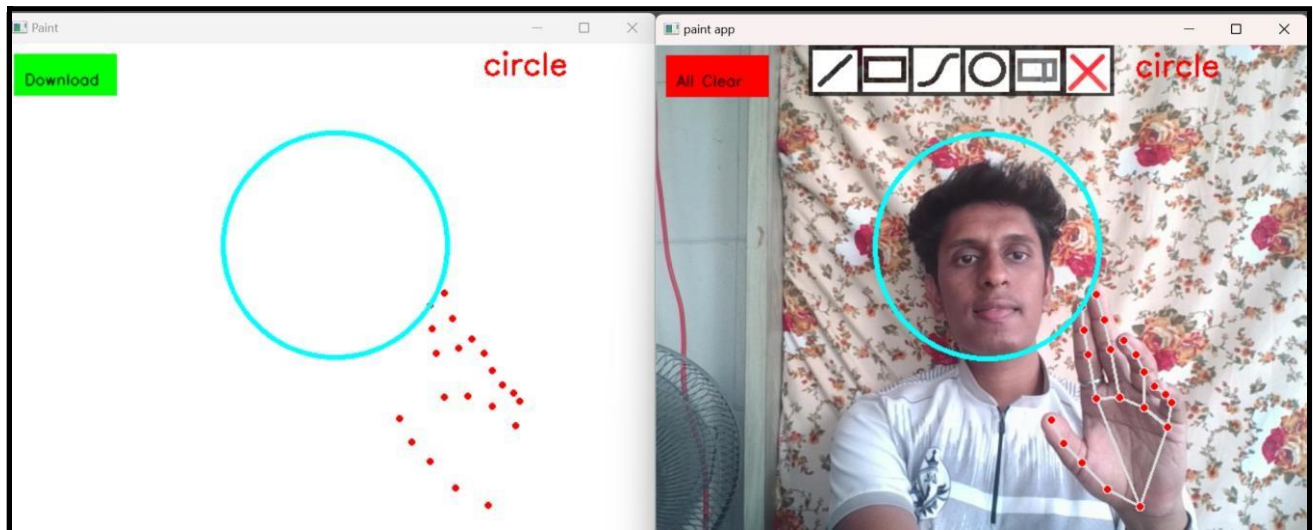


fig.6: Depicting Circle on Canvas

### 6. Exit:

The system provides an exit option where clicking on the cross button closes the OpenCV frame, ending the virtual painting session.

## 6. OBJECTIVE

- To create the mask of color object using contour detection techniques.
- Color Tracking of object at fingertip.
- To make a model for a drawing instrument that utilizes hand signal acknowledgment programming to paint/draw on a Display unit (screen).
- Mapping co-ordinates extracted from hand recognition software to produce a drawing. [5]

## 7. SCOPE OF FUTURE DEVELOPMENT

- To ensure that, the interface is very simple and easily understandable by the user.
- The user should be able to draw what he wishes to draw without any interruptions.
- In future, this is useful for making kids to learn drawing in schools in an interactive way.
- Encompassing artistic expression, education, integrate with augmented reality, annotate PDFs, collaboration, innovation, and user empowerment.

## 8. CALCULATIONS TO FIND THE ACCURACY OF THE MODEL

Methods to find the accuracy of a hand gesture model using 'Ground truth', Gesture Draw and Mouse Draw techniques to draw a square. Where the distance of each point from its expected position is measured to find the error and hence the final accuracy [6].

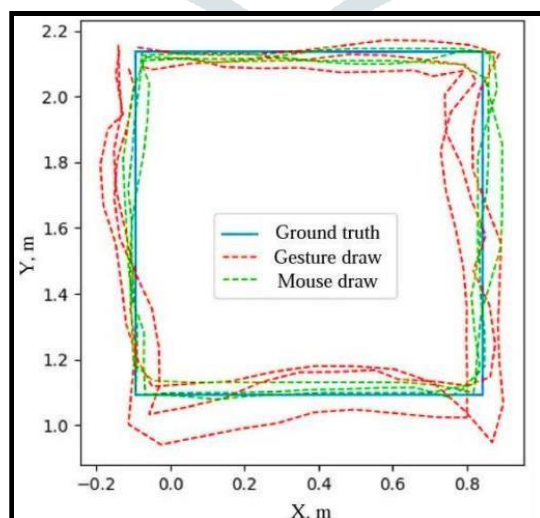


Fig.7: Accuracy of the model

## 9. SYSTEM TESTING

Software testing is a thorough examination conducted to furnish stakeholders with insights into the quality of the product or service being tested. Additionally, it serves as a means to offer an objective and independent perspective on the software, enabling businesses to comprehend and assess the potential risks associated with the implementation of the software. The different test cases for the system are outlined in Table 1.

Table 1: test cases

Sr. No	Description	Input	Expected Value	Actual Value	Result
1	To draw free style on the board with the help of hand	Hand Tracking through webcam	Line of required color is obtained from set of available colors	Line of required color is obtained from set of available colors	PASS
2	To draw a circle shape virtually through Hand on a board.	Hand Tracking through webcam	Circle is obtained with the help of little and index finger	Circle is obtained with the help of little and index finger	PASS
3	To erase the unwanted parts on canvas	Hand Tracking through webcam	The pixels on canvas at index fingertip gets erased	The pixels on canvas at index fingertip gets erased	PASS
4	To select the exit button to terminate through hand	Hand Tracking through webcam	We will be Terminated from the screen to source code	We will be Terminated from the screen to source code	PASS
5	To clear the board with the help of mouse through all clear button	Placing cursor on all clear button using mouse	Board gets cleared i.e. all the shapes drawn are erased	Board gets cleared i.e. all the shapes drawn are erased	PASS
6	Downloading the work done on canvas	Placing cursor on download button using mouse	The work gets saved in images folder	The work gets saved in images folder	PASS

## 10. SNAPSHOTS

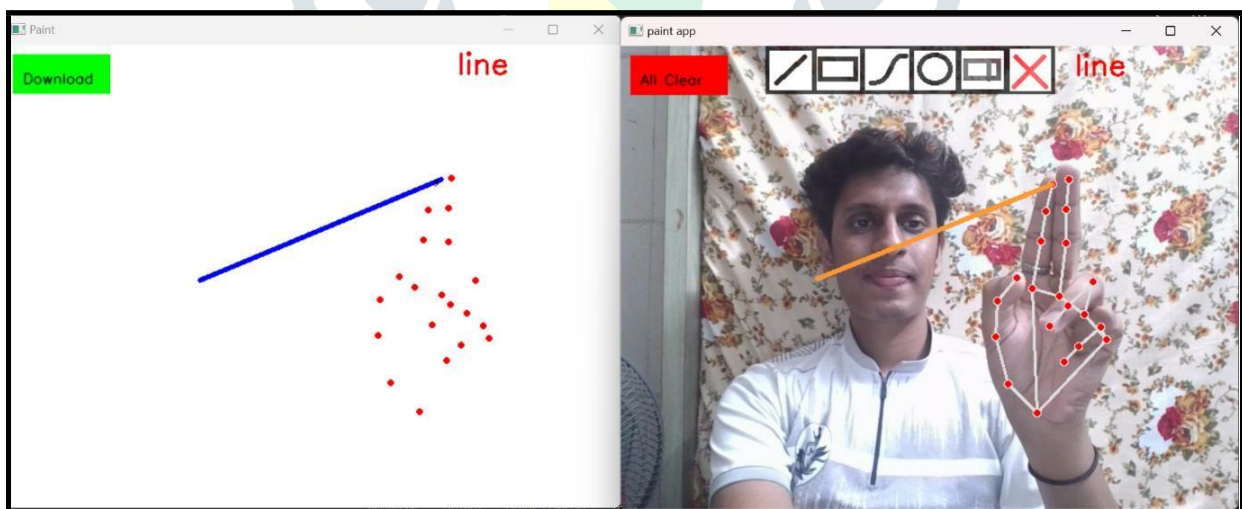


fig.8: draw line



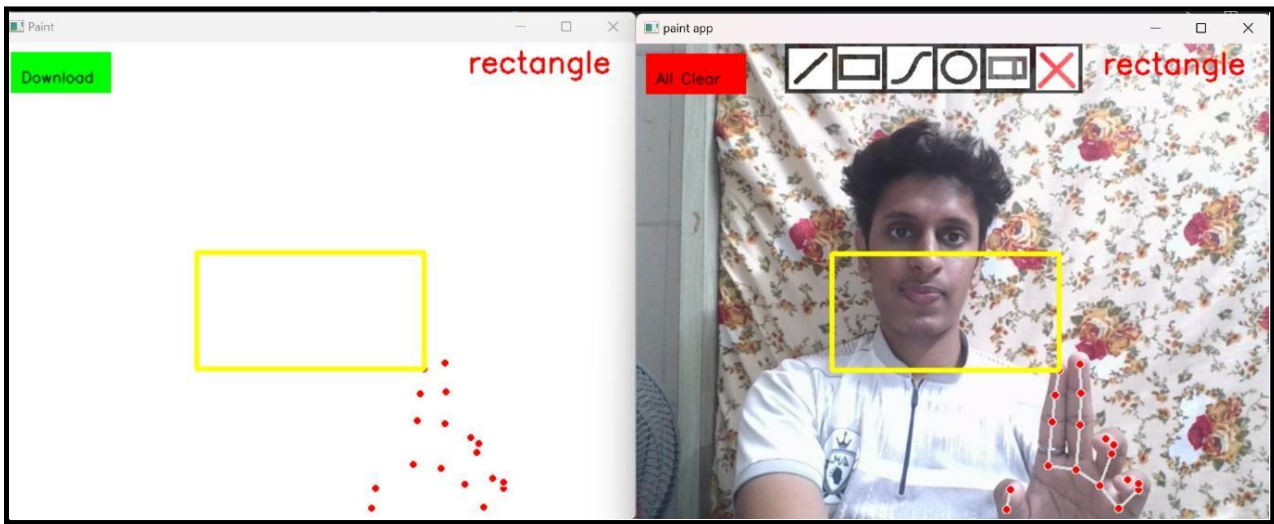


fig.9: draw rectangle

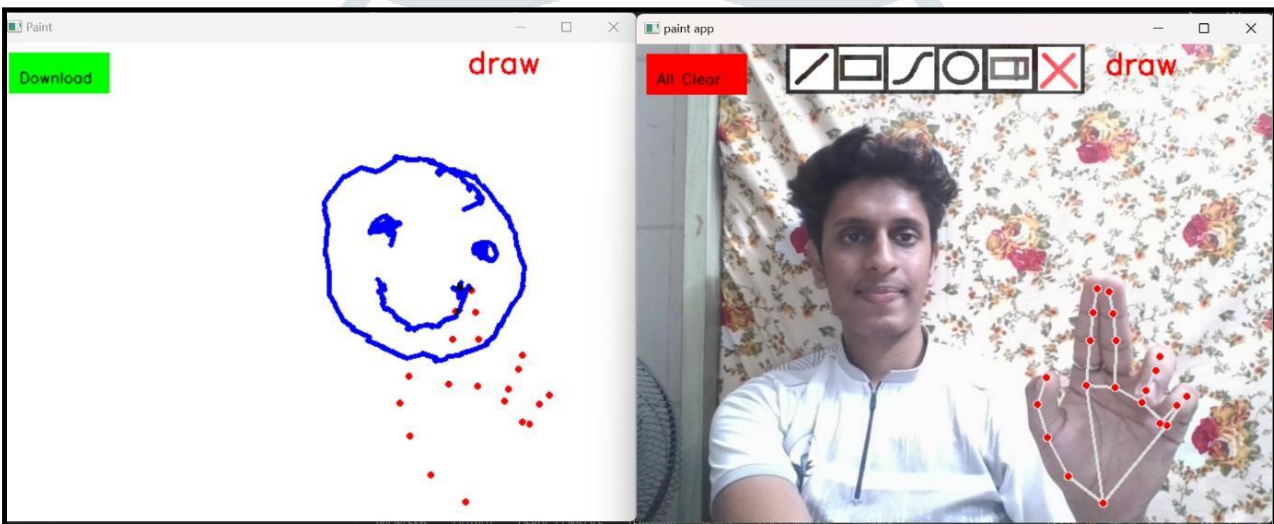


fig.10: free style drawing

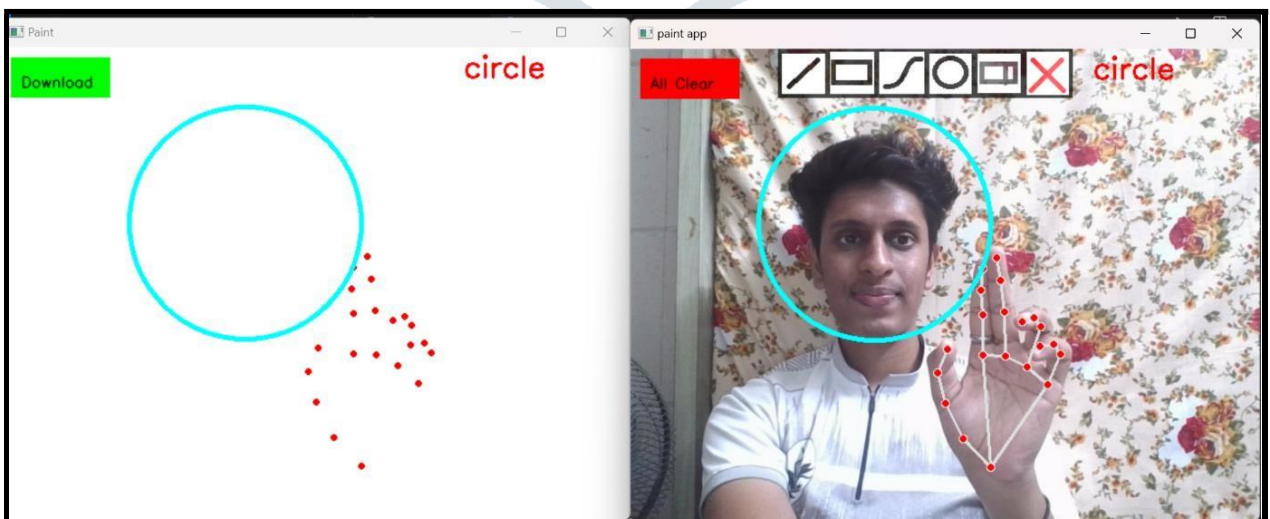


fig.11: draw circle

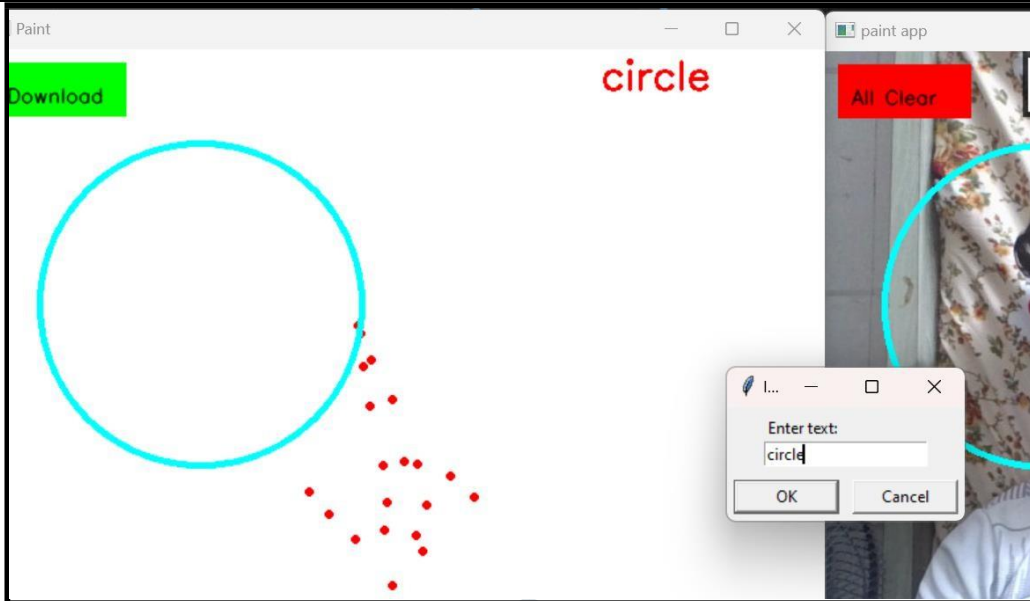


fig.12: add text

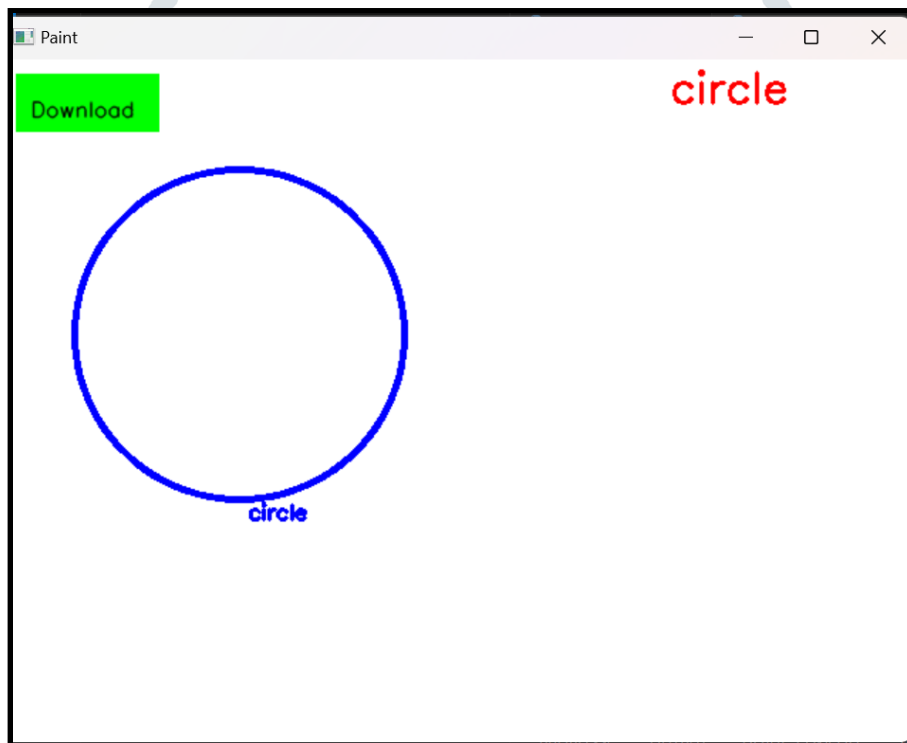


fig.13: circle with labeled

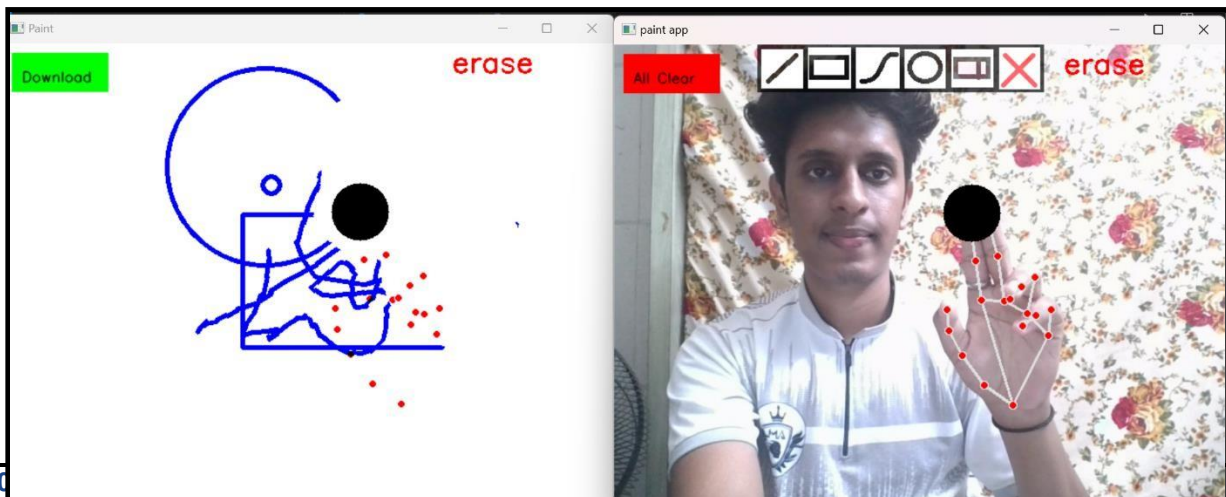


fig.14: unwanted parts erased



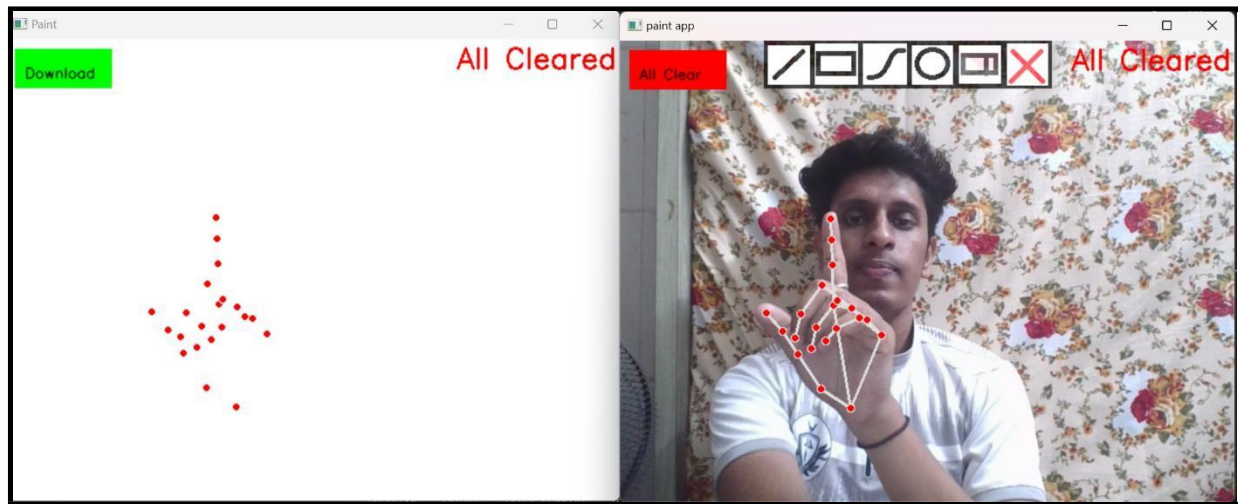


fig.15: all cleared

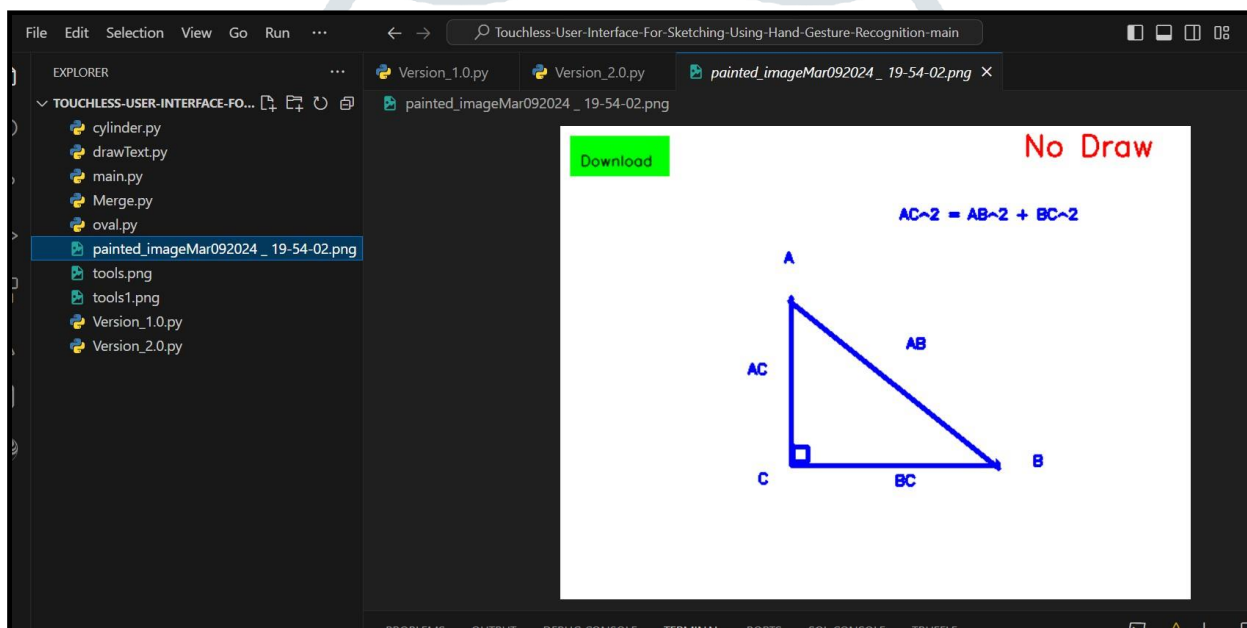


fig.16: Download

## 11. CONCLUSION

In conclusion, the implementation of virtual painting using Air Canvas, Mediapipe, OpenCV, and Python marks a significant advancement in digital artistic expression and interaction. This innovative system provides a hands-free and intuitive drawing canvas, utilizing cutting-edge technologies for an enriched user experience. By leveraging the capabilities of Mediapipe and OpenCV, hand tracking and gesture recognition enable users to draw, annotate PDFs, and manipulate various shapes effortlessly. We [4] used python as, it has many inbuilt libraries and many modules which represent the imagination virtually when used along with OpenCV as well as its morphological processes.

## 12. REFERENCE

- [1] Prof. S. U. Saoji, Nishtha, Dua Akash Kumar Choudhary, Bharat Phogat "AIR CANVAS APPLICATION USING OPENCV AND NUMPY IN PYTHON". Aug 2021
- [2] Dr. B. Esther Sunanda, M. Bhargavi, M. Tulasi Sree, M.R.S. Ananya, N. Kavya "AIR CANVAS USING OPENCV, MEDIAPIPE". May-2022
- [3] Niharika M1, Neha J2, Mamatha Rao3, Vidyashree K P4 "VIRTUAL PAINT APPLICATION USING HAND GESTURES". April 2022
- [4] Pranavi Srungavarapu, Eswar Pavan Maganti, Srilekha Sakhamuri, Sai Pavan Kalyan Veerada, Anuradha Chinta "Virtual Sketch using Open CV". June 2021
- [5] Sayli More1, Prachi Mhatre2, Shruti Pakhare3, Surekha Khot "Air Canvas: Draw in Air". Feb 2022
- [6] Vinit Verulkar , Arnav Waghulde , Nimish Sanghavi , Sumeet Ubale , Prof. Anjali Deshpande "A Survey on Drawing using Hand Gestures for Classroom Learning". January 2024