Gesture Driven Virtual Board

Pratham Gupta, Siddharth Sharma

Project Guide - Neha

Department of Computer Science and Engineering, Galgotias University,

Uttar Pradesh, India

***Abstract—*** ***Hand gesture recognition enables the detection of hand points or coordinates, unlocking new possibilities for human-computer interaction. Advances in computer vision and machine learning are transforming how humans engage with technology. This paper introduces a novel Virtual Whiteboard system that uses hand gestures for interaction, eliminating the reliance on traditional input devices. By employing OpenCV for image processing and MediaPipe for gesture tracking, the system enables users to draw, erase, and manipulate a digital canvas in real time. The proposed approach demonstrates high gesture recognition accuracy and user-friendliness, making it suitable for applications in education, design, and virtual collaboration. Experimental results validate its reliability and potential for widespread use. The system differentiates between drawing and selection modes based on finger gestures—with two fingers (index and middle) raised for drawing mode and only the index finger raised for selection mode. In selection mode, users choose shapes from a predefined list, while in drawing mode, movements in front of the camera are rendered on the screen. This implementation holds promise for situations requiring immediate illustration or explanation.***

***.***

***Keywords— Hand Gesture Recognition, Virtual Whiteboard, Computer Vision, OpenCV, MediaPipe.***

**I. INTRODUCTION**

The evolution of writing has transitioned from traditional methods, like manuscripts and chalkboards, to modern digital forms. The increasing digitization of every aspect of life has also transformed how we express ideas visually. Imagine writing in the air and seeing it appear on a screen—a futuristic concept that is now achievable with advanced technology. This project demonstrates how air writing can bridge traditional practices and digital innovation, setting the stage for a new era of creative and interactive communication.

# II. LITERATURE REVIEW

Artificial Intelligence (AI) enables machines to emulate human thought processes, enhancing efficiency and creating groundbreaking applications. Technologies like OpenCV and MediaPipe—both open-source Python libraries—facilitate processing real-world data for desired outcomes.

Previous research includes:

* **2015**: Development of systems recognizing hand gesture for robot control or communication.
* **2015**: Image-processing algorithms for gesture-based mouse and cursor control.
* **2021**: Gesture datasets were created from video frames, but accuracy remained challenging.
* **2016**: Finger tracking via LEDs and webcams for character recognition.

These studies underline the growing interest in gesture recognition while highlighting areas for improvement, such as accuracy and real-time usability.

# III. EXISTING SYSTEM

Current gesture recognition systems primarily focus on aiding communication for deaf and mute individuals by converting hand gestures into characters. While effective, these systems often face challenges such as being time-consuming and having limited real-life applications.

**3.1 Drawbacks:**

* High time consumption.
* Limited usability in dynamic environments

# IV. PROPOSED SYSTEM

The proposed Virtual Whiteboard extracts hand coordinates using a webcam and processes these points for shape selection and drawing on a digital screen. The screen is divided into multiple segments based on (x, y) coordinates, allowing precise control for shape selection and drawing. Once a shape is selected, users can draw freehand or create structured shapes. This system’s flexibility makes it suitable for diverse applications, from education to real-time design.

**Advantages:**

* Reduced processing time.
* High accuracy in gesture recognition.
* Intuitive interaction through a virtual board.
* Practical real-world applications.

.

# V. MODULES & IMPLEMENTATION

# MODULE 1: SET THE FRAME

The system initializes by displaying shape options at the top of the screen. Shapes can be selected when the user’s hand moves above a predefined threshold. Below the threshold, the drawing mode is activated. The screen is divided into regions corresponding to selectable shapes, enabling intuitive interaction

# SOURCE

cap = cv2.VideoCapture(0)

while True:

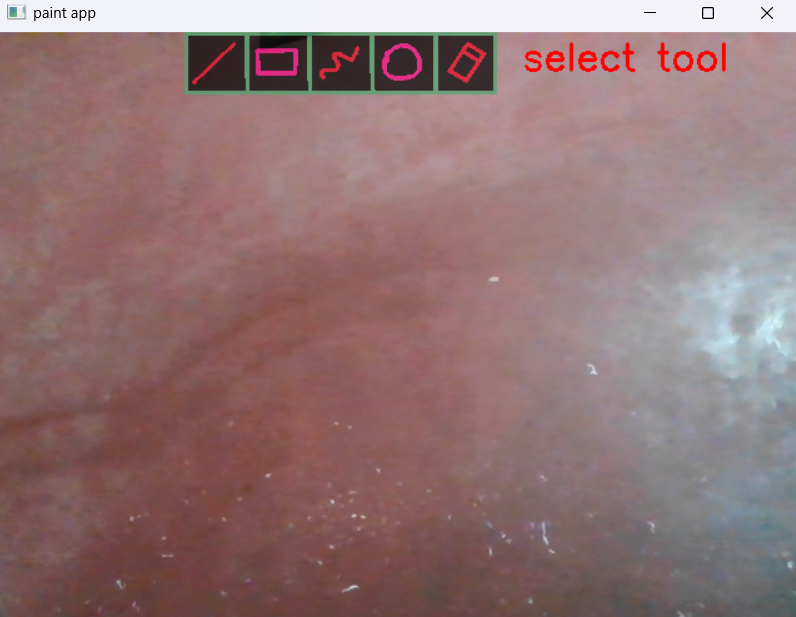
\_, frame = cap.read()

frame = cv2.flip(frame, 1)

rgb = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

output = hand\_landmark.process(rgb)

**RESULT**



# MODULE 2: RECOGNIZE HAND POINTS

Hand landmarks are detected using MediaPipe. The system identifies key points such as fingertips and calculates their coordinates to determine the user’s intent.

# SOURCE

    if op.multi\_hand\_landmarks:

        for i in op.multi\_hand\_landmarks:

            draw.draw\_landmarks(frm, i, hands.HAND\_CONNECTIONS)

            x, y = int(i.landmark[8].x\*640), int(i.landmark[8].y\*480)

    if x\_index < max\_x and y\_index < max\_y and x > ml:

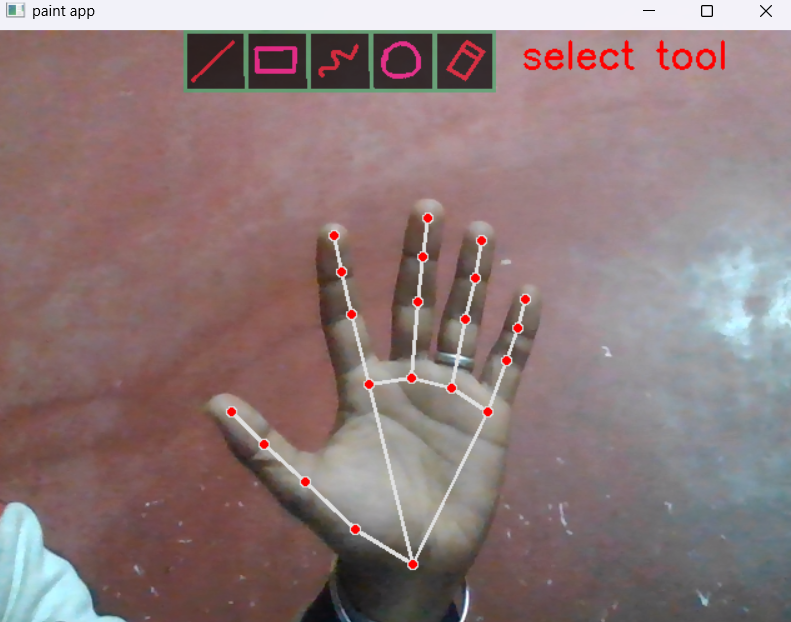
                if time\_init:

                    ctime = time.time()

                    time\_init = False

                ptime = time.time()

# RESULT



# MODULE 3: POSITION THE POINTERS

Pointers are used for selection and drawing. In selection mode (index and middle fingers raised), a rectangle appears between the fingers to guide the user. Drawing mode activates when only the index finger is raised.

# SELECTION MODE

The screen is segmented into boundaries for each shape. Moving the pointer within a boundary selects the corresponding shape. A default shape is assigned initially, and users can switch shapes as needed.

**SOURCE**

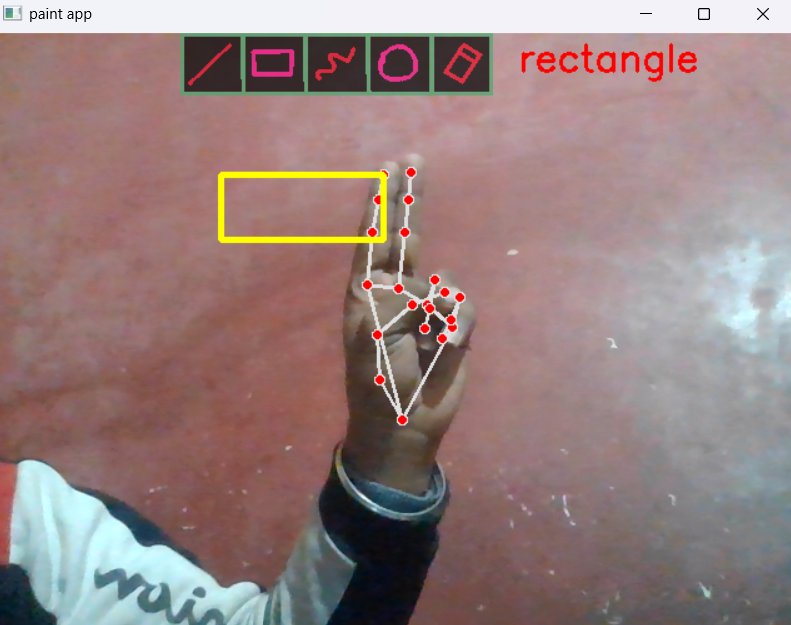
if index\_raised(yi, y9):

if not(var\_inits):

xii, yii = x, y

v2.rectangle(frm, (xii, yii), (x, y), (0,255,255), thick)

# RESULT



# MODULE 4: SELECT SHAPES

The screen is segmented into boundaries for each shape. Moving the pointer within a boundary selects the corresponding shape. A default shape is assigned initially, and users can switch shapes as needed..

# SOURCE

**SOURCE**

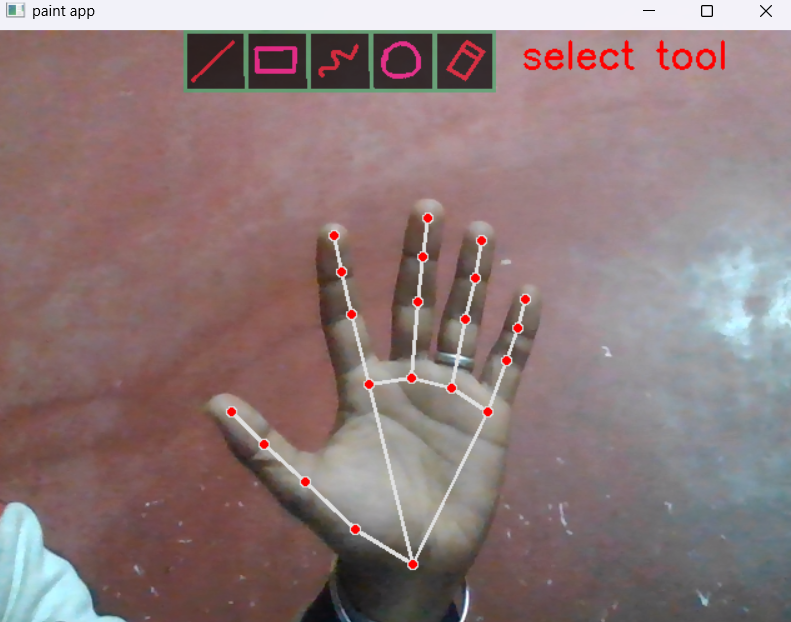
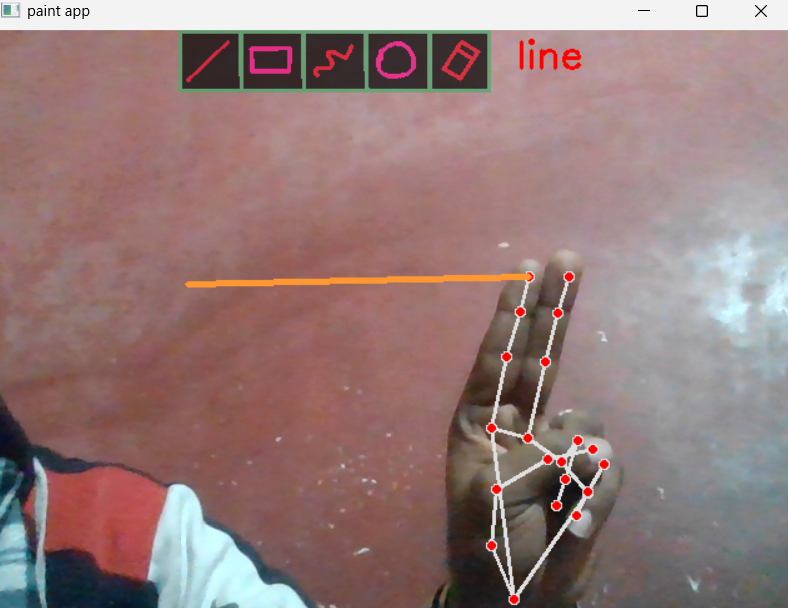
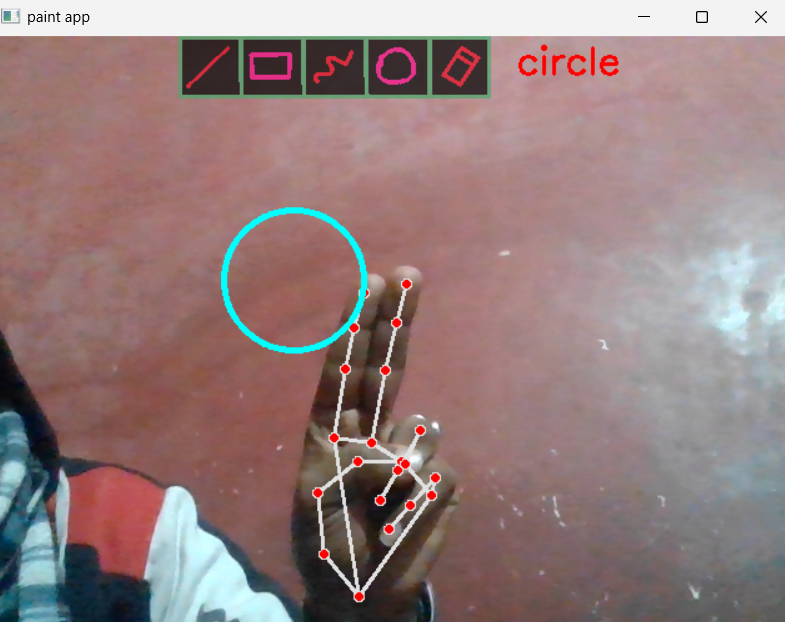
xi, yi = int(i.landmark[12].x\*640), nt(i.landmark[12].y\*480)

 cv2.circle(frm, (xii, yii), int(((xii-x)\*\*2 + (yii-y)\*\*2)\*\*0.5), (255,255,0), thick)

                y9  = int(i.landmark[9].y\*480)

:

**RESULT**



**Drawing Mode**

Drawing mode allows users to draw or write whatever they want to display on the screen. This mode is activated when the index finger is raised, which is visually indicated by a circle around the finger. The drawing starts with a default color and size.

The screen is divided into sections, each representing a specific shape. By moving the pointer within a section, users can select the corresponding shape. A default

shape is pre-selected, but users can switch to other shapes as needed.

# if x < max\_x and y < max\_y and x > ml:

# if time\_init:

# ctime = time.time()

# time\_init = False

# ptime = time.time()

# cv2.circle(frm, (x, y), rad, (0,255,255), 2)

# rad -= 1

# if (ptime - ctime) > 0.8:

# curr\_tool = getTool(x)

All the values adapt based on the screen configuration ensuring flexibility and responsiveness.

**MODULE 5: START PAINTING**

This is the place for what the whole process is made for. Now whatever the user writes on the air in front of the webcam will be shown on the screen. This is done by using

the OpenCV python library and its built-in functions

**SOURCE**

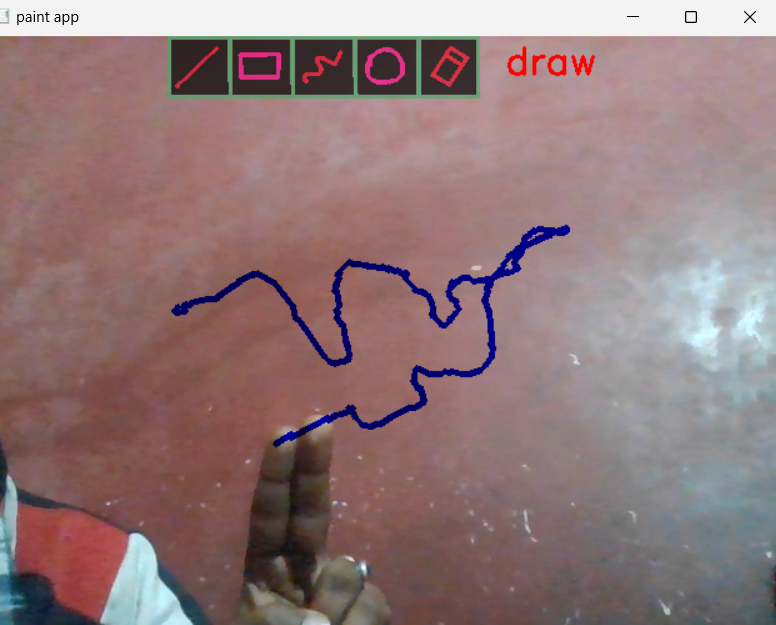
xi, yi = int(i.landmark[12].x\*640), int(i.landmark[12].y\*480)

y9  = int(i.landmark[9].y\*480)

cv2.line(mask, (prevx, prevy), (x, y), 0, thick)

  prevx, prevy = x, y

# RESULT



# VI. CONCLUSION AND FUTURE WORK

The hand gesture-based Virtual Whiteboard showcases the potential of computer vision for intuitive, touch-free interaction. By integrating OpenCV and MediaPipe, the system provides real-time performance suitable for various domains. Future enhancements will focus on:

1. Improving robustness in challenging environments, such as poor lighting or busy backgrounds.

2. Introducing multimodal features, such as voice integration.

3. Scaling the system for multi-user interactions and cross-platform compatibility.

4. Providing advanced customization tools, including layers and shape-drawing modes.

# REFERENCES

1. Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 370-377, 2016.
2. P. Ramasamy, G. Prabhu, and R. Srinivasan, "An economical air writing system is converting finger movements to text using a web camera," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, pp. 1-6, 2016.
3. Saira Beg, M. Fahad Khan and Faisal Baig, "Text Writing in Air," Journal of Information Display Volume 14, Issue 4, 2013.
4. Alper Yilmaz, Omar Javed, Mubarak Shah, "Object Tracking: A Survey", ACM Computer Survey. Vol. 38, Issue. 4, Article 13, Pp. 1-45, 2006.
5. Pranavi Srungavarapu, Eswar Pavan Maganti, Srilekkha Sakhamuri, Sai PavanKalyan Veerada, Anuradha Chinta, VirtualSketch using Open CV, ISSN: 2278- 3075 (Online), Volume-10 Issue-8, June 2021
6. Pranavi Srungavarapu, Eswar Pavan Maganti, Srilekkha Sakhamuri, Sai PavanKalyan Veerada, Anuradha Chinta, VirtualSketch using Open CV, ISSN: 2278- 3075 (Online), Volume-10 Issue-8, June 2021
7. First International Conference on Computer Engineering International Journal of Scientific Research in Science and Technology Virtual Painting with Opencv Using Python Yash Patil1, Mihir Paun1, Deep Paun1, Karunesh Singh1, Vishal Kisan Borate Volume 5 Issue 8, November-December-2020.
8. Suryansh Pratap Singh a, Akshat Mittal b, Manas Gupta c, Soumalya Ghosh d, Anupam Lakhanpale Comparing Various Tracking Algorithms In OpenCV Vol.12 No.6 (2021), 5193-5198
9. Bhumika Nandwana, Satyanarayan Tazi, Sheifalee Trivedi Dinesh Kumar khunteta, Santosh Kumar vipparthi, A Survey Paper on Hand Gesture Recognition
10. Hemalatha Vadlamudi, Evaluation of Object Tracking System using Open-CV In Python International Journal of Engineering Reseаrch & Technology (IJERT)Vol. 9 Issue 09, September-2020
11. Siddharth Mandgi1, Shubham Ghatge1, Mangesh Khairnar1, Kunаl Gurnani1, Prof. Amit Hаtekar2 Object Detection And Tracking Using Image Processing Vol. 8, Issue 2, ( Part -1) February 2018, pp.39-41
12. S. Soo 2014 Object detection using Haar-cascade

Classifier (Inst. Comput. Sci. Univ. Tartu) vol. 2 no. 3 pp 1–12