Virtual Painter

1st Aayush Sanghvi
College of Engineering
Northeastern University
sanghvi.aay@northeastern.edu

2nd Yogi Shah
College of Engineering
Northeastern University
shah.yo@northeastern.edu

Abstract: **Virtual Painter is an innovative application that transforms the way users interact with digital content by integrating hand gesture recognition technology with real-time drawing capabilities. Utilizing the robust frameworks of OpenCV and MediaPipe, this application offers users an intuitive and immersive experience, allowing them to draw, erase, take screenshot, clear contents on the canvas and customize their digital creations directly onto live video feeds. With the ability to recognize various hand gestures, users can effortlessly select drawing tools, adjust settings such as brush sizes and colors, and manipulate their canvas in a fluid and natural manner. The application features distinct modes, including draw mode for adding artistic elements, erase mode for removing unwanted content, and settings mode for adjusting the tip size, eraser size and, choose colors of their choice. Additionally, users have the option to capture screenshots of their artwork and clear the canvas to start fresh. Virtual Painter empowers users to unleash their creativity, explore new artistic possibilities, and engage with digital content in a dynamic and personalized way. Whether for artistic expression, educational purposes, or recreational enjoyment, Virtual Painter offers a versatile and immersive platform for users to bring their imagination to life.**

Keywords—OpenCV, MediaPipe, Hand gestures, styling.

I. INTRODUCTION

The “Virtual Painter” project focuses on using computer vision algorithms like OpenCV and MediaPipe to detect and interpret hand gestures in real time. This technology enables users to select drawing tools, colours, and brush sizes to create 2D drawings directly on a video feed, revolutionizing the way users interact with live video feeds by combining hand gesture recognition and real-time drawing capabilities. The project's functionality is divided into four distinct modes based on the number of fingers detected: Hover Mode for finger hovering, Erase Mode for object erasing, Draw Mode for drawing objects, and Setting Mode for changing pen tip size and opacity.

Additionally, the project allows users to draw on a canvas using their index fingertip and enhances the user experience by providing a more comprehensive set of features. The functionality of the "Virtual Painter" project relies on OpenCV and MediaPipe libraries to identify distinct hand features in real time, enabling different functions based on these features. The project introduces four distinct modes: Hover Mode, which allows finger hovering across the video stream to start drawing; Erase Mode, for erasing unwanted objects within the video feed or the complete board; Draw Mode, enabling users to draw objects directly onto the live video feed using the tip of their index finger; and Setting Mode, allowing users to change the pen tip size, eraser, and the opacity of the pen. The project's innovative approach aims to provide a fun and intuitive way for users to express their creativity and interact with digital content in a unique and engaging manner, ultimately revolutionizing the user experience with live video feeds.

II. RELATED WORK

A. Painting with Hand Gestures using MediaPipe

The document discusses the implementation of a painting application using hand gestures, facilitated by the MediaPipe framework and OpenCV library. The project aims to leverage hand gesture recognition for applications such as industrial automation control, sign rehabilitation for individuals with physical disabilities, virtual environments, and human-computer interaction. The significance of gestures in human communication and their potential for non-contact interaction with machines is emphasized. The application involves recognizing the hand and fingertip movements using the MediaPipe framework, capturing real-time hand motions, and utilizing OpenCV for computer vision to enable the user to draw on a canvas by tracking the fingertip movements of the index finger. The system allows the user to draw, select colours, and clear the canvas using intuitive hand gestures, ultimately creating a virtual painting experience.

The document delves into the related work, highlighting various research papers and their contributions to the field. It discusses the use of the OpenCV library for image and video processing, as well as the MediaPipe framework for low-latency performance and time-series data synchronization. The MediaPipe framework is described as modular, reusable, and fast, offering pre-built machine learning solutions and support for TensorFlow and TF Lite inference engines. Moreover, the application's algorithm and libraries are detailed, emphasizing the use of MediaPipe for hand gesture identification and tracking, along with OpenCV for computer vision. The proposed system's architecture, including the hand and fingertip recognition process, is illustrated.

The results presented demonstrate the successful implementation of the painting application, showcasing its ability to detect live video of a person, recognize the index fingertip, and enable drawing and painting activities using freehand gestures. The images provided depict the empty canvas, color selection, painting or writing, and pen-up and pen-down actions, showcasing the practical functionality of the application. Additionally, the document offers a comprehensive list of references, spanning various research papers and publications in the domain of hand gesture

recognition, sign language recognition, and human-computer interaction.

In conclusion, the document outlines the successful development of an AI-based application for painting using hand gestures, underscoring its potential for enhancing human-computer interaction and providing an intuitive and engaging experience for users. The document provides a holistic overview of the project, from its objectives and related work to the technical implementation and demonstrated results, offering valuable insights into the intersection of human-computer interaction, computer vision, and gesture recognition.

B. Virtual AI Painter using OpenCV and MediaPipe

This research paper explores the development of an application using OpenCV and MediaPipe to enable real-time drawing and communication through hand gestures. The paper discusses the use of Python as the programming language and emphasizes the significance of OpenCV, a library for computer vision, and MediaPipe, an open-source framework for media processing. The application allows users to draw by capturing hand motion and interact with a computer or webcam. It also enables palm detection, color tracking, and text recognition for real-time gesture control and communication.

The system's design involves hand pose detection, hand region segmentation, localization of hand centroid, and fingertip detection model using Deep Learning algorithms. The paper provides a detailed account of the experimental results, showcasing the practical application of the virtual AI painter, and concludes with the potential implications of the application in various fields. The study references several related works and previous theories, highlighting the relevance of the proposed system in the context of gesture-controlled computer systems, air writing systems, and object tracking. Furthermore, the paper presents the detailed implementation steps, including setting up the paint interface, locating interest contours, and initiating video frame capture. The application's GUI and the workflow of the system are also depicted, demonstrating the practicality and user-friendliness of the developed virtual AI painter.

Finally, the research paper concludes by emphasizing the interactive and reciprocal nature of the application, its potential applications in education, and the seamless drawing experience it offers through hand gestures and color selection. Overall, the study provides a comprehensive overview of the development, implementation, and potential impact of the virtual AI painter using OpenCV and MediaPipe.

C. Virtual Painter using Artificial Intelligence and OpenCV

The document discusses the development of a virtual painter application using artificial intelligence (AI) and OpenCV. It emphasizes the integration of AI as a tool for artistic expression, highlighting that AI cannot create art without human intent. The document outlines the functionalities of OpenCV, a library for image processing, which includes tasks such as face and shape detection, text recognition, and augmented reality app development. Additionally, it introduces MediaPipe, a framework for media processing with applications like face detection, multi-hand tracking, and object detection and tracking. The existing systems for displaying data, including keyboard, speech-to-text, and touchscreen methods, are briefly discussed, citing their advantages and limitations. Furthermore, the proposed system's components and functionalities are detailed, including hand pose detection, hand region segmentation, background subtraction, hand centroid localization, and drawing line using the position of the contour. The conclusion emphasizes the wide applicability of the virtual painter application in various fields and underscores its efficiency and user-friendly interface for drawing and conveying thoughts.

In essence, the document presents the advancements in AI and computer vision technology for creating a virtual painter application that enables users to draw using hand movements in front of a webcam. It underscores the integration of AI as a tool for artistic expression, while also providing a comprehensive overview of OpenCV and MediaPipe functionalities. The discussion on existing and proposed systems offers a detailed insight into the technical aspects of the virtual painter application, including hand pose detection, hand region segmentation, background subtraction, hand centroid localization, and drawing line using the position of the contour. The conclusion emphasizes the efficiency and user-friendly nature of the application, highlighting its potential applications in various fields and its contribution to human-computer interaction.

III. FUNCTIONALITY

The project harnesses the capabilities of OpenCV and MediaPipe libraries to enable real-time identification of distinct hand features, facilitating seamless interaction with live video feeds. These features serve as the foundation for a range of functionalities, organized into four distinct modes, each activated by the detection of a different number of fingers:

A. Hover Mode

Users can hover their finger across the video stream to indicate where they want to begin drawing. This mode provides a convenient starting point for creating artwork or annotations.

Fig. 1. Hovering with Index finger as a pointer

B. Draw Mode

With this mode activated, users can draw objects directly onto the live video feed using the tip of their index finger. This

intuitive feature allows for effortless creation of artwork or annotations in real time.

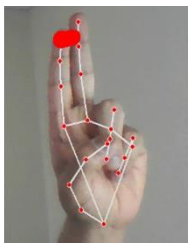

Fig. 2. Drawing with Two fingers(Index + Middle Fingers)

C. *Erase Mode*

Users have the ability to erase unwanted objects within the video feed or clear the entire canvas space. Whether correcting mistakes or starting fresh, this mode offers precise control over editing the content on the screen.

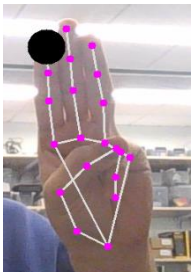

Fig. 3. Eraser with three fingers(Index + Middle + Ring fingers)

D. *Setting Mode*

This mode provides users with the flexibility to customize their drawing experience. Users can adjust the tip size of the pen or eraser, as well as modify the opacity of the pen to achieve the desired visual effect.

Fig. 4. Setting menu with different option of color, and sizes

E. *Screenshot Mode*

Users can capture screenshots of their artwork with a simple gesture, allowing them to save their creations locally for future reference or sharing with others.

Fig. 5. 6 fingers gesture to automatically take Screenshot.

F. *Clear All Mode*

This mode clears the entire canvas space, providing users with a clean slate to start anew. It offers a quick and convenient way to reset the canvas and begin fresh creative endeavours.

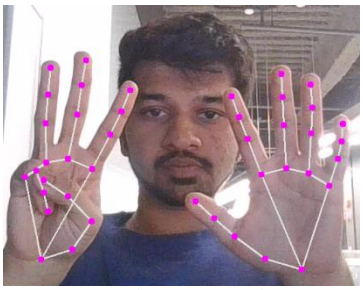

Fig. 6. 8 finger/ second hand 3 finger gesture for clear screen.

By offering these diverse modes of interaction, the project aims to empower users to express their creativity and interact with digital content in a dynamic and engaging manner.

IV. Methodology

A. *Initialization*

Application imports all the necessary library like time, HandTracker and NumPy and predefined variables like the drawing color, eraser color, brush size, eraser size are set with a default value with a zero-value image canvas of 720*1280 pixels.

B. *Looping*

The Video capturing is set and with every frame findHands() method detects the hands and gives a count with its landmarks. With the findPosition() method we get the position of this landmarks on the image and get a list with Landmark id, x, y values.

C. *Gesture Recognition*

Now we know the tips of all the fingers in the frame. By using the fingerUp() method receive the number of finger that are up or down in a 0 or 1 value format and by proper if condition a gesture is recognized which is then asked to perform different tasks in every frame interval.

D. *Two Hand Interaction*

For this type of setup, we worked on implementing a settings menu for the availability of choice between the size of the brush, eraser and color options to paint with. As two more addons we implemented Complete board clearing and Screenshot capturing ability to not only take the image with the drawings but to have images with just the drawings on a black background. The detection and gesture were recognized in a way that no matter what is happening the second hand the 3 basic functionalities namely drawing, erase and hovering was not interrupted.

E. *Canvas Processing*

For now putting all these drawing and erasing from the drawings is performed but processing the canvas image with the help of thresholding, imgInv which converts all the drawings a white and background as black. This image is then implemented on the frame with bitwise operations and get the

drawings and their respective color on the frame. This canvas image is stored and on ever frame based on the response from the use it keeps adding or subtracting to the canvas.

F. Display and Userinteraction

The image is now displayed on the screen and checked if the user has flagged a gesture to save the image or not. If it is flagged then the images are stored with timestamp as its name with a jpg file format.

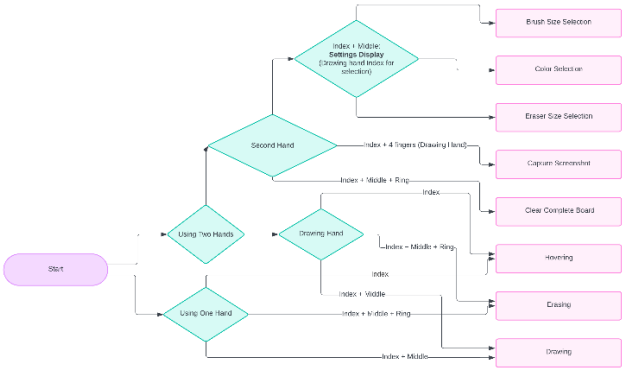

Fig. 7. Flowchart for complete program states

V. Results and Analysis

Following figures are how the system responds to the user interaction.

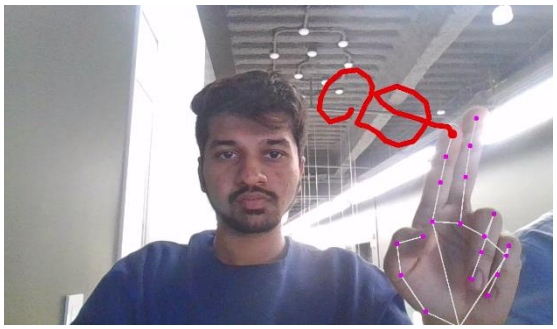

Fig. 8. Drawing with one hand

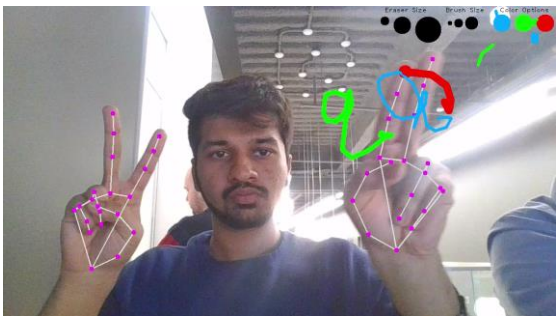

Fig. 9. Second hand Settings gesture and drawing with other

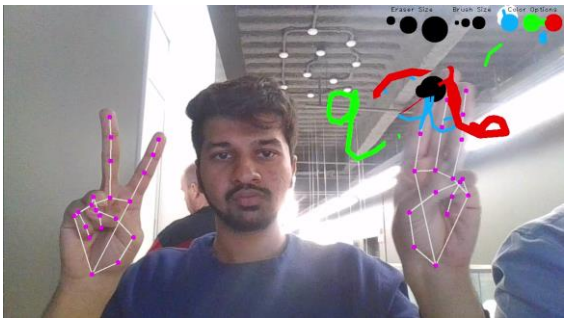

Fig. 10. Erasing with different sized erasers

In the figure 9 you can see the drawings are done with different color and different sizes while keeping the settings menu on.

Two of the major drawback we have with this system is that as the settings menu is on the right it is difficult for the left handed people to operate and perform the Settings operation and that is because when you try to reach a corner opposite to your hand you basically overlap the hands during your travel and the system then forgets which hand is drawing hand and which is second hand and it interchanges the operations of the hands. And the system acts poorly in very bright lighting conditions.

VI. Acknowledgment

From all the team members it is thank Prof. Bruce Maxwell for guiding through all the concepts of Computer vision giving a intuitive ability to think about any vision problem. Thank you to the teaching assistants for countless hours of back and fourth troubleshooting with simple small problems to complex issues.

VII. References

[1] Tiwari, Satyam & Vartak, Deep & Verma, Vivek & Sayyed, Anasali. (2023). Virtual Hand Gesture Painting.

[2] R.Vasavi; Nenavath Rahul; A Snigdha; K. Jeffery Moses; S.Vishal Simha. (Volume. 7 Issue. 12, December - 2022) "Painting with Hand Gestures using MediaPipe." , International Journal of Innovative Science and Research Technology (IJISRT), www.ijisrt.com. ISSN - 2456-2165 , PP :- 1285-1291. https://doi.org/10.5281/zenodo.7514430

[3] Rajaji, Shanthi. (2024). Virtual Painting through Hand Gestures: A Machine Learning Approach.

[4] Volume 5, Issue 1, January 2015 ISSN: 2277128X, International Journal of AdvancedResearch in Computer Science and Software Engineering : Research Paper – Gesture Controlled Computer.

[5] https://mediapipe.readthedocs.io/en/latest/solutions/hands.html