A VIRTUAL PAINTER

Samruddhi Sawant
Department of Computer Engineering
Vasantdada Patil College of Engineering & Visual Arts
Mumbai, India
vu1f2122002@pvppcoe.ac.in

Diksha Waghmare
Department of Computer Engineering
Vasantdada Patil College of Engineering & Visual Arts
Mumbai, India
vu1f2122014@pvppcoe.ac.in

Yukta Jadhav
Department of Computer Engineering
Vasantdada Patil College of Engineering & Visual Arts
Mumbai, India
vu1f2122033@pvppcoe.ac.in

Aviral Shukla
Department of Computer Engineering
Vasantdada Patil College of Engineering & Visual Arts
Mumbai, India
vu1f2122061@pvppcoe.ac.in

Guide Name
Prof. Atul Shintre
Department of Computer Engineering
Vasantdada Patil College of Engineering & Visual Arts
Mumbai, India

Abstract— **In today's world it is really important to make digital tools more interactive and engaging. The "virtual painter" integrated with Artificial Intelligence is an intelligent application that uses hand gestures to perform creative and fun tasks. It has developed with Artificial Intelligence and frameworks like MediaPipe and OpenCV. The system allows users to draw in front of camera by their hand gestures, available shapes and resize using gestures, and even control ppt slides without touching anything. This user-friendly application has enhanced human-computer interaction, making creative arts and presentations easier and more fun.**

Keywords— MediaPipe, OpenCv, Human-Computer-interaction.

I. Introduction

In the rapid growth of technology, human interaction has become more natural, interactive and accessible. The "gesture art" is an imaginative and participatory application that applies Artificial Intelligence and gesture detection to improve user familiarity. Hence project enables users to interact with canvas in natural way, making activities like drawing, resizing shapes, and controlling presentation hands-free.

The application includes three modules: Module that identifies and decodes anything written or drawn on virtual canvas, a virtual shape drawing module that allows users to draw and resize available shapes using hand gestures and third module is presentation command module that allows effortless navigation of PowerPoint slides using gestures.

This virtual Painting application is constructed with powerful frameworks such as MediaPipe and OpenCV. This System ensures correct gesture detection and immediate response. The project illustrates the potential of integrating Artificial Intelligence and gesture-based interactions to generate practical applications for creativity, education and professional presentations, enhancing human-computer interactions[7].

II. Literature Survey

The Gesture Art project represents a significant improvement in digital interaction, merging the power of computer vision, real-time hand tracking and artistic creativity to transform the way users engage with digital canvases. Various existing studies have opened the door for advancements in gesture-based interaction, each focusing on unique aspects of hand gesture recognition and digital painting. Unlike these systems, the models frequently lack precision, usability, and real-life applicability. This project overcomes these limitations by using MediaPipe for advanced hand tracking and OpenCV for real-time image processing. Traditional methods, such as those that depend on LED-equipped fingers or split video frame datasets, have low accuracy and lack flexibility, making them impractical for creative and professional use. These constraints limit the application of gesture recognition in more dynamic and interactive environments. In comparison, our system utilizes a precision-based approach, mapping screen space into a grid of (x, y) coordinates to enable effective color selection and air drawing.

R. Shanthi, S. Abdul Azeez, and N. Deepak Kumar's study 'Virtual Painting through Hand Gestures: A Machine Learning Approach' concentrates on using MediaPipe and OpenCV to create a painting tool powered by hand gestures. This paper highlights the potential of integrating computer vision frameworks to use seamless and precise drawing in real-time. The authors highlight the use of gesture recognition algorithms to track hand and finger movements, which closely corresponds with the core functionalities of our AI Virtual Painter.

Building on previous studies, our system introduces AI-powered tools that track hand movements and allow for more advanced drawing and interactive features. By combining gesture recognition with machine learning, we improve the user experience, making it ideal for artistic creation as well as practical uses like presentations and educational materials. In the paper "Painting with Hand Gestures using MediaPipe," R. Vasavi and colleagues explore digital painting using MediaPipe for tracking and OpenCV for processing images. They focus on using colour markers on fingertips to recognize hand gestures accurately. Our project builds on this by adding AI features that can predict and suggest design elements, offering more than just basic gesture tracking. Additionally, the paper "Gesture Recognition is an Emerging Technology Field" discusses gesture-based interactions without the need for traditional devices, which aligns with the goals of our AI Virtual Painter. Many existing systems, such as those described in "Gesture Recognition: Techniques and Applications," face challenges like poor accuracy and limited real-world use, relying on old hardware or algorithms. This makes them less practical for creative and professional work. Our AI Virtual Painter overcomes these problems by using MediaPipe and OpenCV to ensure better accuracy and speed, making gesture recognition more useful for a wider variety of tasks.

One of the key improvements of our project is scalability, as it merges creative artistic expression with practical applications in education, presentations, and interactive learning. The System's adaptive feature ensures broad accessibility, making it usable for individuals with different levels of mobility. Additionally, the integration of machine learning models allows the system to predict and suggest design elements in real-time, boosting creativity and enabling more complex digital artwork. Our system goes beyond traditional painting by integrating the capability to handle PPT presentations. This functionality allows users to interact with presentation slides using hand gestures, making it ideal for educational and professional environments. The AI-driven tools not only track hand movements for drawing but also provide gesture-based control for slide navigation, eliminating the need for separate hardware. This feature enhances the usability of our system, making it a versatile tool for both artistic expression and efficient presentation management.

III. IMPLEMENTATION AND METHODOLOG

The virtual Painter is a piece of AI software that uses computer vision, deep learning, natural language processing to interact with the user via his/her hand gestures. The Ai system runs on OpenCV for video capture and processing in real-time, providing high-level interaction between the user and the virtual canvas. For building efficient hand tracking, Mediapipe Hands is applied among the technologies. It detects and tracks 21 landmarks on the hand, by which it can determine the fingertip positions and make the difference between the drawing and erasing gesture, The application has also picked up streamlit, a python-based web framework, that gives the user an opportunity to participate in the process not only with live video but also with Ai-generated results.

Deque (collections. Deque) is a data structure that is used to store the user's drawing points and is responsible for the smooth strokes and real-time rendering. The index finger is raised when the system goes into drawing mode, and it stops when the thumb and index finger come close together. A button "ERASE" is also included in the design of the virtual canvas, which when pressed, changes the overlay from the canvas colour to black, effectively removing the strokes of the drawings. Moreover, NumPy is employed for image manipulation while using the image matrix, and it is ensuring that the optimized processing of video frames is applied.

The application's most distinctive characteristic is it's using Google Gemini API which allows the content that the user has just drawn to analyse by the powerful generative AI model in a very useful way. When the user draws on the canvas, the image is converted from OpenCV format to PIL (Python Imaging Library) format itself and next, it is sent to the Gemini AI model for more processing. The AI acts based on the image content and might do image-filling, text-recognition, and object-finding, besides math problem-solving. In a case where the model identifies a math expression, it also can be very clever and solve it. In the case of textual sketches, the system is able to take out the information and also to summarize. For the sketch of a rough object, the AI defines the intended subject and gives a refined interpretation[8].

For a better output, the device presents the virtual image to the real one using the camera and processing it with cv2.addWeighted(), where the canvas is overlaid onto the video feed with some perfect transparency. Moving on, also the cv2.bitwise_and() and cv2.bitwise_or() functions are used to implant drawing digitally with a real-time video. The program contains an exception handling mechanism that ensures that errors related to webcam access or AI processing will not affect the user experience. In addition, the usage of the graceful interruption by the user with KeyboardInterrupt allows easy exit without causing a crash.

In a nutshell, the Virtual Painter creatively blends computer vision, AI-generated image analysis, and web-based interaction, thus producing an entirely new artistic digital environment where users can do sketching. The combination of hand tracking MediaPipe, AI-driven content recognition Gemini API, and real-time rendering OpenCV & Streamlit gives the users a dynamic and interactive experience. The technology is an example of a harmonious blend of gesture recognition, artificial intelligence, and real-time video processing which makes it a full-fledged solution for virtual sketching and AI-assisted drawing applications.

MediaPipe hands provides 21 specific landmarks that help in detecting and recognizing hand gestures. These landmarks are structured to track different parts of the hand, enabling accurate gesture-based interaction such as drawing, clicking, or erasing the virtual painter.

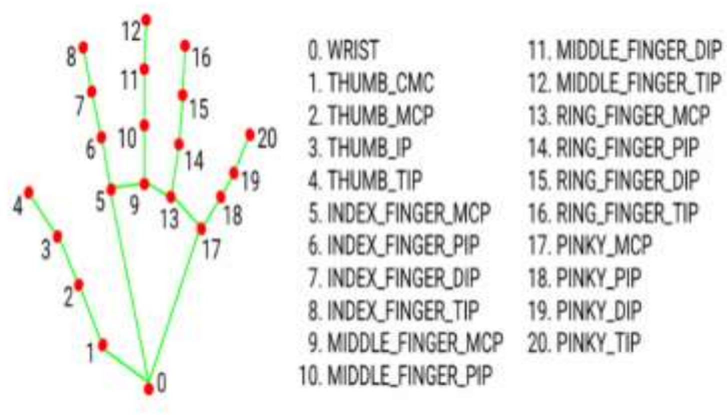


Fig 1: Hand Landmarks

Landmark Numbering and Their Positions

Each landmark corresponds to a unique location on the hand. The numbering follows a hierarchical structure from the wrist to the fingertips.

Landmark ID	Name	Description	Used in Gesture Recognition
0	Wrist	Base of the hand, acts as a reference point.	Stabilizes hand tracking.
1	Thumb CMC	First joint of the thumb, near the palm.	Helps in thumb movement detection.
2	Thumb MCP	Second joint of the thumb.	Used for pinch detection.
3	Thumb IP	Third joint of the thumb.	Helps in recognizing closed fists.
4	Thumb Tip	Fingertip of the thumb.	Important for gestures like pinching.
5	Index MCP	First joint of the index finger.	Helps in detecting pointing gestures.
6	Index PIP	Second joint of the index finger.	Used in gesture tracking.
7	Index DIP	Third joint of the index finger.	Helps in detecting bent fingers.
8	Index Tip	Fingertip of the index finger.	Important for drawing actions.
9	Middle MCP	First joint of the middle finger.	Stabilizes the hand in tracking.
10	Middle PIP	Second joint of the middle finger.	Helps in hand pose detection.
11	Middle DIP	Third joint of the middle finger.	Used in grasp detection.
12	Middle Tip	Fingertip of the middle finger.	Important for two-finger gestures.
13	Ring MCP	First joint of the ring finger.	Used in complex gestures.
14	Ring PIP	Second joint of the ring finger.	Helps in tracking multiple fingers.
15	Ring DIP	Third joint of the ring finger.	Used in bent finger detection.
16	Ring Tip	Fingertip of the ring finger.	Less used in simple gestures.
17	Pinky MCP	First joint of the pinky (smallest finger).	Stabilizes the hand in recognition.
18	Pinky PIP	Second joint of the pinky.	Helps in detecting small hand gestures.
19	Pinky DIP	Third joint of the pinky.	Used in complex interactions.
20	Pinky Tip	Fingertip of the pinky.	Useful for precise gestures.

Rules for Gesture Recognition Using Landmarks

Finger State Detection It is not easy to detect the state of a finger on a mobile device but it may be done by calculating the degree of rotation of the thumb and the first/last knuckle and then comparing to a threshold as the thumb rotates the most when being used as the touch pointer. If the tip of a finger is above the previous joint, it is considered open (extended).

If the tip is below or aligned with the previous joint, it is considered closed (bent).This is used to determine which fingers are narrowed or unfolded.

Drawing Mode Activation: In the drawing mode, if just the index finger (landmark 8) is pointed while the rest are folded, then the system goes to the drawing mode. The system keeps a record of the fingertip coordinates of the detected index finger as the finger moves back and forth to draw. If only the index finger (landmark 8) is extended while others are folded, the system enters drawing mode. The system locates the index fingertip to keep track of drawing movement.

Erasing Mode Activation : If both index (8) and middle fingers (12) are extended at the same time, it triggers the erase mode. The drawing is canceled by the system as the user raises both index finger and the middle finger up. If both index (8) and middle fingers (12) are extended together, it triggers the erase mode. The system considers this as a signal to wipe out some areas of the whiteboard.

Stopping a Gesture : The pinch gesture is sent when the thumb (4) and index finger (8) are getting near each other and thus the system quits the drawing action.

Gesture for AI Analysis In the case of two fingers (index and middle) being folded and the others being extended, the system actively records the drawing and forwards it to AI-based analysis.

Click or Selection Gesture When the index finger (8) touches the middle of the screen, it becomes suitable for use to select an option.

How These Landmarks Work in Virtual Painter:

- The index fingertip (8) is the main control point for drawing and selection.
- Thumb position (4) is used to detection of a stop of the touch to delete drawing.
- The middle finger (12) tends to be additionally powerful in multi-finger gestures like erasing.
- The wrist (0) provides a stable angle for the hand tracking to[o]l to be customized.

Utilizing the 21 landmarks and sticking to the rules above, the Virtual Painter gets very exact gesture gestures that a user has to make in order to draw, delete and analyze the input image with only hand movements.

The system is a video-sharing service that takes video from a webcam, processes the frames to detect hand landmarks, and interprets the movements of the user to do certain tasks like drawing with a freehand, line, rectangle, and circle creation, as well as an eraser tool for modifications.

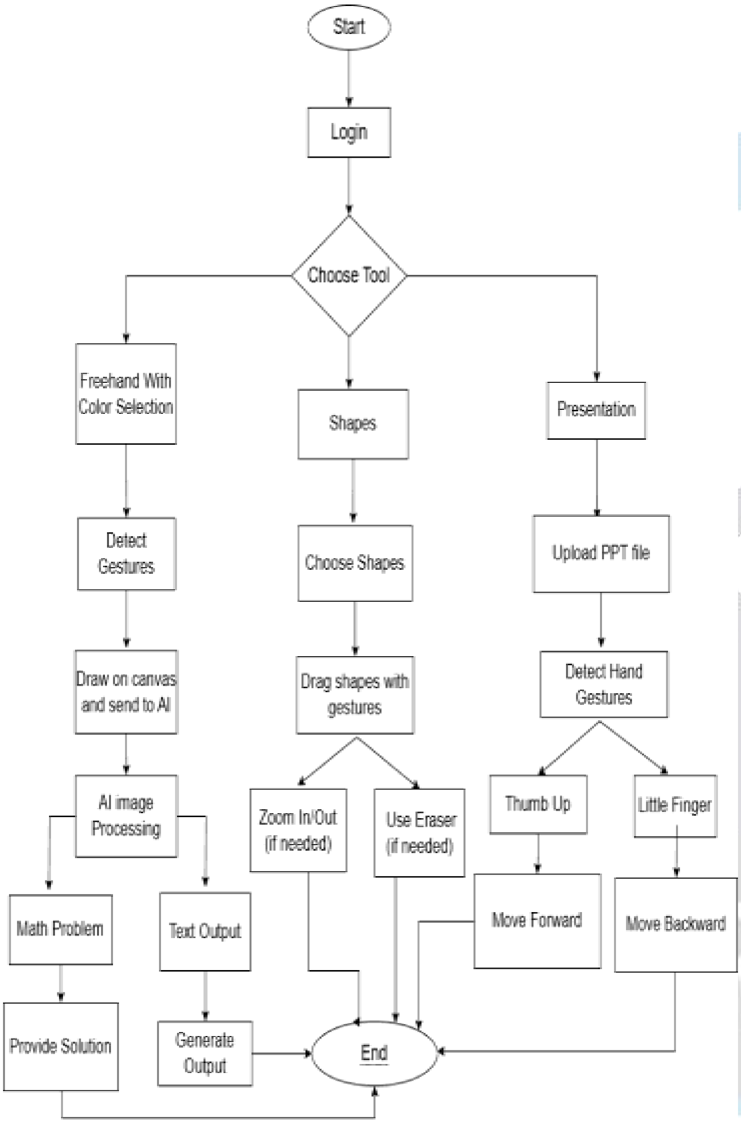
Firstly, the importation of indispensable libraries is done, including MediaPipe, which is responsible for fast hand tracking and OpenCV, which facilitates frame handling, image processing as well as GUI interactions. NumPy is a library that provides an interface for optimizing mathematical operations. Binary mask is the thing which provides the ability to superimpose the drawn content onto the video feed. In the meantime, timing restrictions are regulated by the use of the time module which is operated.

Am the MediaPipe Hand Model is initialized with a minimum detection confidence of 0.6 and tracking confidence of 0.6. In this way, precise landmark tracking of hands can happen in all environments even the dynamic ones. MediaPipe drawing utilities are helpful in seeing hand landmarks and connections at the same time.

An image appears in the OpenCV window and it is different from other drawings tools. There are OpenCV's cv2.imread() which offers the user masterpiece to draw and allows him to choose a tool by the cursor location. getTool(x) maps the x-coordinate of the index finger tip to the correct tool, enabling a tool to be selected seamlessly. The index_raised(yi, y9) function that detects if the index finger is up or not is a control mechanism for the drawing actions.

the main actions is to horizontally flip each frame (cv2.flip(frm, 1)) so as the user could navigate easily. The frames are converted to the RGB format, as the MediaPipe Hands model uses this format, and after that, the hands are detected. The tip of the index finger (landmark 8) is employed to distinguish the tools while other landmarks (landmarks 9 y 12) give feedback for the drawing actions as well.

In the case tool is already selected different actions are performed by gesture recognition to represent gestures



drawn

Fig.2 Flow chart Of Hand Gesture Painting.

on the screen. The system uses OpenCV drawing functions such as cv2.line(), cv2.rectangle(), and cv2.circle() to make the dynamic shapes of the mask as it is later combined with the video feed using cv2.bitwise_and(). The mask ensures that drawings stay on different frames allowing the user to have a good quality experience.

While drawing by hand, the system always monitors the index finger's movement and then uses the consecutive points to draw the lines. For lines, rectangles, and circles, the system figures out when the user lifts one's through the r erroring that as the start point and lowering the finger as end point will look like a tab. The eraser tool operates in such a way that the index finger lifted status is captured and then the content is taken out from the drawing mask through the white circles present at the positions being detected.

Last but not least, the GUI tools of OpenCV (cv2.putText() and cv2.imshow()) are used to show the current tool and to update the live interface. The process is carried out until the user presses the Escape (ESC) key if the user wants to close the OpenCV windows and release the webcam capture which is done by cv2.destroyAllWindows() and cap.release().

This system has the ability to effectively include the real-time hand tracking, gesture recognition, and digital drawing functionalities into an interactive application that is to demonstrate the use of computer vision techniques for human-computer interaction (HCI)[7]. The approach can be applied to virtual whiteboards, AR-based drawing applications, and accessibility tools for users with mobility impairments and many more innovative ideas.

This framework does not need a camera if it is set up in a way that is contingent on you holding your palm up to the screen, as you are tracking hand gestures by the reflection on the screen.

detectorHand.findHands(img): A video will be taken every 32 ms to detect hands and the coordinates of the landmark points of each hand will be obtained.

detectorHand.fingersUp(hand): These are functions that measure the distance of each finger to the centerline based on length and classification of stillness and then classify the hand gesture.

With the new feature, It is easy for users to extend the functionality of existing systems by including gestures that are performed in the vicinity of the device camera and that need the hand to move in specific directions and to perform specific gestures. The MediaPipe framework is capable of processing data that is recorded in real-time, meaning it can be used in areas including but not limited to, hand-drawn painting or presentation control, which demands accurate gesture tracking.

Additionally, to track the hands the HandTrackingModule of cvzone moves the hands from frames to the list of coordinates:

cv2.VideoCapture(0): Executes the camera so that the human eye can capture what is happening.

cv2.flip(img, 1): Moves the video feed horizontally to make sure the interaction is more natural.

cv2.cvtColor(img, cv2.COLOR_BGR2RGB): Colors of the frames are changed from BGR (OpenCV default) to RGB, because MediaPipe needs them in RGB for the purpose of finding hands[3].

cv2.line(): it's code, which creates a dynamic shape with the help of a function line() call, such as a line for gesture recognition or a gaze threshold for direction recognition.

cv2.circle(): A method to visualize interaction points and draw annotations.

cv2.rectangle(): A technique to draw rectangle shapes during the gesture interaction.

cv2.imshow(): This frames appear with the result of the processed frames. On the one hand, the video is the source from the webcam, and on the other video annotation layers, on the visual display, in real-time.

cv2.bitwise_and(): This will take the frame that is output as a mask for the drawing operation of a given video to the video feed which is the movie. The reason for this is that we want to present separate frames with and without the drawings.

cv2.putText(): Shows the name of the tool being processed on the screen.

ul>This OpenCV functions are crucial to interacting with the live video feed, rendering graphical materials like lines, shapes, the annotation and giving you a clear visual experience.

Using cvzone for the hand gesture tracking and OpenCV for image processing, the system can be provided with a touchless, interactive experience with the gestures controlling the dynamic drawing tools. By this method the movements of the hand are thought with accuracy, and the user can operate perfectly the drawing strokes, which is the right solution for realtime purposes, such as digital painting, virtual whiteboards or presentations.

IV. CONCLUSION AND FUTURE SCOPE

This virtual paint application transforms digital innovation and productivity providing AI-driven platform that improves user interaction with computers. The application provides users with features to draw on the screen control the presentation and choose various tools and shapes to create innovative designs. By utilizing hand gestures recognition, it offers a seamless, gesture based for completing tasks, enhancing both productivity and user experience.

While the system gives a remarkable potential, there are some limitations, such as high computational costs and inaccurate gesture recognition due to camera quality. Environmental factors can affect the performance

ACKNOWLEDGMENT

We would like to express our sincere appreciation to Dr. Atul Shintre for his invaluable support and guidance during this study.

REFERENCES

[1]. Ayesha Gurnani, Viraj Mavani, and Vandit Gajjar – "Hand gesture real time paint tool-box: Machine learning approach", 2018, DOI: 10.1109/ICPCSI.2017.8391833 – IEEE Publisher\

[2]. Chen, X., Chen, Y., & Zhang, L. (2019). A Novel Virtual Painting System Based on Hand Gesture Recognition. In 2019 International Conference on Mechatronics and Control Engineering (ICMCE) (pp. 434-439). IEEE.

[3]. Mazumdar, Dharani; Talukdar, Anjan Kumar; Sarma, Kandarpa Kumar. (2013). [IEEE 2013 1st International Conference on Emerging Trends and Applications in Computer Science (ICETACS) - Shillong, India (2013.09.13-2013.09.14)] 2013 1st International Conference on Emerging Trends and Applications in Computer Science - Gloved and free hand tracking based hand gesture recognition., (), 197–202. doi:10.1109/ICETACS.2013.6691422

[4]. Chen, Qing, Nicolas D. Georgians, and Emil M. Petriu. "Real-time vision-based hand gesture recognition using hear-like features." In Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE, pp. 1-6. IEEE, 2007

[5]. Fernando, Matheesha; Wijaya nayaka, Janaka . (2013). [IEEE 2013 IEEE 8th International Conference on Industrial and Information Systems (ICIIS) - Peradeniya, Sri Lanka (2013.12.17-2013.12.20)] 2013 IEEE 8th International Conference on Industrial and Information Systems - Low cost approach for real time sign language recognition.,(),637642. doi:10.1109/iciinfs.2013.67 32059R.

[6]. Author(s), "Deep learning-based fast hand gesture recognition using representative frames," in Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA), Nov. 2016. doi: 10.1109/DICTA.2016.7797030.

[7]. Suree Pumrin and Thittaporn Ganokratanaa - "The Vision-Based Hand Gesture Recognition Using Blob Analysis", April 2017, DOI: 10.1109/ICDAMT.2017.7904987 - IEEE Publisher

[8]. Yewale, Shweta. K.; Bharne, Pankaj. K. . (2011). [IEEE 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETZNCC 2011) - Udaipur (2011.04.22-2011.04.24)] 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC) - Hand gesture recognition using different algorithms based on artificial neural network. , (), 287–292. doi:10.1109/ETNCC.2011.6255906