AIR CANVAS USING COMPUTER VISION AND MEDIAPIPE

Mr.D.Ramana Kumar [1]
CSE, Anurag University
Hyderabad, India
ramanad74@gmail.com

Abhinav Takkada [2]
CSE, Anurag University
Hyderabad, Telangana
abhinavtakkada@gmail.com

Thummala Mounika [3]
CSE, Anurag University
Hyderabad, Telangana
thummalamounika2003@gmail.com

Vannekala Charan Teja [4]
CSE, Anurag University
Hyderabad, India
vannekalacharanteja@gmail.com

Kolthuru Aditya [5]
CSE, Anurag University
Hyderabad, India
adityakolthuru@gmail.com

Abstract. In the realm of technological progress, the integration of human gestures as a means of virtual control has gained prominence. Project Air Canvas stands as a testament to this evolution, focusing on the development of a motion-to-textual converter that redefines the act of drawing through hand gestures. This project is rooted in the refinement of hand tracking systems, leveraging Open Computer Vision Library (OpenCV) and Mediapipe technologies. The core objective revolves around transforming hand moments into an intuitive drawing tool, empowering users to create various shapes and erase effortlessly by merely waving their hand. Distinguishing itself from conventional methods that often involve complexity and time-intensive processes, Project Air Canvas aims to streamline and enhance user experiences. Through innovative technologies and simplified methodologies, the project utilizes system cameras to track hand gestures, enabling seamless drawing.

***Keywords*:** OpenCV, MediaPipe

1. INTRODUCTION

In recent years, the evolution of human-computer interaction (HCI) has seen a remarkable shift towards more intuitive and immersive interfaces. One such innovation gaining traction is the concept of air canvases, enabling users to draw and interact with digital content using hand gestures in the air. This paradigm shift is fueled by advancements in computer vision and machine learning technologies, particularly MediaPipe and OpenCV, which have paved the way for real-time hand gesture recognition and fingertip detection [1][7].

The concept of air canvases builds upon previous research in gesture-based interaction systems, such as bare finger 3D air-touch systems and vision-based hand tracking systems, which have explored novel ways to interact with digital interfaces without physical tools [3]. By leveraging Deep Learning techniques and sophisticated algorithms, researchers have endeavored to create intuitive and responsive air canvas systems that enhance user creativity and productivity [4].

The development of air canvas systems holds immense potential across various domains, from digital art and design to online teaching and collaborative workspaces [9]. With the proliferation of web-based applications and smart devices equipped with cameras, the accessibility and usability of air canvas technology are poised for exponential growth [7].

We aim to contribute to the burgeoning field of air canvases by presenting a comprehensive overview of the underlying technologies and methodologies employed in their development [8]. Drawing insights from a diverse range of research articles and conference papers, we delve into the intricacies of hand gesture recognition, fingertip detection, and real-time interaction mechanisms, shedding light on the challenges and opportunities in this exciting area of HCI [12].

Through a synthesis of existing literature and empirical research, we endeavor to provide valuable insights and recommendations for the design and implementation of air canvas systems, paving the way for more intuitive and immersive digital experiences in the future.

2. LITERATURE SURVEY

1. Tracking Advancements:

Papers like (Bare Finger 3D Air-Touch System Using an Embedded Optical Sensor Array for Mobile Displays) [3] and (Tracking of Flexible Brush Tip on Real Canvas: Silhouette-Based and Deep Ensemble Network-Based Approaches) [4] explored tracking techniques using cameras or sensors. Recent works might delve into more sophisticated approaches like:

• 3D Hand Pose Estimation: Enabling control over brush depth and pressure in the virtual canvas.

2. Beyond Drawing:

While the core concept revolves around drawing, papers like [11] (Aerosculpt: A Symphony of Creativity on the

Air Canvas- Revolutionizing Digital Art and Design through Intuitive Hand Gestures) with "Aero sculpt" hint at broader applications.
We might see research on:
•3D Sculpting: Manipulating virtual objects in the air using hand gestures.
3. User Experience and Integration:

The focus might shift towards user experience, as seen in papers like (Air Doodle: " A Realtime Virtual Drawing Tool.) [5] with "Air Doodle." Here's what future research could explore:
•Multi-touch and Collaboration: Enabling multiple users to interact with the Air Canvas simultaneously.
•Integration with Existing Tools: Allowing seamless transfer of Air Canvas creations to design or editing software.

4. Hardware and Accessibility:
Papers like (Air Canvas: Draw in Air) [1] (Air Doodle :" A Realtime Virtual Drawing Tool") might inspire research on hardware advancements.
•Standalone Air Canvas Devices: Dedicated hardware for a more immersive experience.
•Integration with Mobile Devices: Utilizing smartphone cameras for on-the-go Air Canvas creation.
5. Educational and Assistive Applications:
Beyond artistic expression, Air Canvas could find applications in:
•Education: Interactive learning experiences through air-based manipulation of virtual objects.
• Assistive Technologies: Enabling individuals with limited mobility to interact with computers using hand gestures.

Table 1 : Literature Survey

S.No	Author(s)/Title	Advantages	Disadvantages
1	Sayil More, Parachi Mhatre, Shruti Pakhare, Surekha Khot. Air Canvas: Draw in Air [1].	1. The system enables hands-free drawing, allowing users to interact with the digital canvas using gestures rather than physical tools. 2. The system's modular design, with distinct components for color tracking, contour detection, and frame processing, offers flexibility and ease of maintenance. It allows developers to update or replace individual modules without affecting the overall functionality of the system.	1. Drawing in the air using hand gestures may lack the precision and control of traditional drawing tools like pens or styluses. This can result in less accurate or detailed artwork, particularly for complex designs or fine details. 2. The system's reliance on specific hardware components, such as a webcam and display unit, may limit its accessibility and usability.
2	B Anand, T.Vinod, ,M Srinivas Rao, Interaction through Computer Vision Air Canvas [2].	1. The air canvas system allows users to draw and interact with the digital canvas without the need for physical contact with a device. By simply waving their finger in the air, users can create drawings or write text. 2. The system can be set up with various hardware configurations, including built-in or external web cameras, making it adaptable to different environments and user preferences.	1. Drawing in the air using finger waving may lack the precision and control of traditional drawing tools, leading to less accurate or detailed artwork. Users may find it challenging to create intricate designs or fine details due to the inherent limitations of the interaction method. 2. Setting up the system and calibrating it for accurate color detection and tracking may require technical expertise and time-consuming adjustments.
3	Guo-Zhen Wang, Yi-Pai Huang, Tian-Sheeran Chang, and TsuHan Chen, "Bare Finger 3D Air-	1. The proposed 3D virtual-touch system offers a compact and portable solution for interacting with	1. The depth range of the system is currently limited to up to 3 cm, which may restrict its applicability for

	Touch System Using an Embedded Optical Sensor Array for Mobile Displays” [3].	3D images on mobile displays. By embedding optical sensors and angular scanning illuminators into the display pixels and edges, respectively, the system can be integrated seamlessly into mobile devices without requiring additional bulky hardware. 2. Unlike camera-based systems, which rely on high-resolution images for 3D positioning, this system operates without the need for a camera. By using embedded optical sensors and IR backlighting, it can accurately determine the 3-axis (x, y, z) position of a fingertip above the display surface, offering a more efficient and reliable interaction method.	certain interactions requiring greater depth sensitivity. 2.Multi-touch functionality may be limited by occlusion effects, where overlapping fingertips or objects obstruct each other's visibility to the sensors. This can result in inaccuracies or failures in detecting multiple touch points, particularly in scenarios with complex interactions or overlapping gestures.
4	Joolekha Bibi Joolee; Ahsan Raza; Muhammad Abdullah; Seokhee Jeon, Tracking of Flexible Brush Tip on Real Canvas: Silhouette-Based and Deep Ensemble Network-Based Approaches [4].	1. The deep ensemble network-based approach offers accurate tracking of the brush tip position on the canvas. By utilizing a combination of LSTM Autoencoder and 1-D CNN, the network captures complex relationships between the 3D pose of the brush handle and the 2D brush tip position, allowing for precise estimation during drawing sessions. 2. Unlike the silhouette-based approach, which may require specially aligned frames and cameras, the deep ensemble network approach reduces the complexity of the system setup.	1. Training a deep ensemble network requires a significant amount of data and computational resources. Collecting and preprocessing data for network training can be time-consuming, and training the network itself may require powerful hardware and substantial training time to achieve optimal performance. 2. Implementing and maintaining a deep ensemble network-based system may introduce complexity, especially for users who are not familiar with deep learning techniques.
5	Soham Pardeshi, Madhuvanti Apar , Chaitanya Khot , Atharv Deshmukh, Air Doodle :“ A Realtime Virtual Drawing Tool” [5].	1. The Air Doodle application offers a user-friendly interface that allows users to interact with the system using a solid-coloured cap for drawing and navigation. This intuitive interaction method enhances user experience and makes the application accessible to a wider audience, including those who may not be familiar with complex input devices or software interfaces.	1. The effectiveness of the color detection algorithm and overall performance of the Air Doodle application heavily rely on the surrounding environment, including factors such as lighting conditions and the color and size of the object used for interaction. 2. Implementing features such as color detection, gesture recognition, and

		2. With features such as writing/drawing, color change, screenshot capture, and screen clearing, the Air Doodle application provides users with versatile functionality for creative expression and practical use	real-time interaction requires robust technical solutions and careful consideration of algorithm performance and computational resources.
6	Melvin Cabatuan , Isaiah Jassen Tupal . Vision-Based Hand Tracking System Development for Non-Face-to-Face Interaction [6].	1. The developed system allows users to draw and erase on a digital canvas in real-time using an object tracked by the vision-based hand tracking system. This enables fluid and natural interaction with the digital drawing environment, enhancing user creativity and productivity. 2. The program offers convenient export options, allowing users to save their drawings as image files or record their drawing process as a video file. This feature facilitates sharing of drawings with others or creating instructional videos for educational purposes.	1. The color thresholding method used for object tracking may be prone to errors when there are objects in the background with similar colors to the tracked object. 2. While the MediaPipe Hands tracking solution offers an alternative method for hand tracking, the accuracy of handwriting tracing may be limited compared to direct object tracking.
7	Shaurya Gulati, Ashish Kumar Rastogi, Raghav Pradhan, Chetan Gupta, Mayank Virmani, Rahul Jana, Paint / Writing Application through WebCam using MediaPipe and OpenCV [7].	The system enhances accessibility for individuals with disabilities, such as those with hearing impairments or difficulties using traditional keyboards. It provides an intuitive and effortless way to write and communicate, allowing users to express themselves more effectively and independently.	The effectiveness of the system may be influenced by environmental factors such as lighting conditions and background clutter. Variations in lighting or the presence of objects with similar colors to the tracked hand could lead to inaccuracies in hand gesture recognition and writing, affecting the system's reliability.
8	Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review [8].	1. HMMs provide a probabilistic framework for modeling gesture dynamics, enabling uncertainty quantification in gesture recognition. 2. HMMs inherently incorporate DTW, allowing for alignment of gesture sequences with different durations, which is crucial for accurate recognition.	1. HMMs typically assume a first-order Markov property, which may not fully capture the complexity of some gesture sequences that exhibit higher-order dependencies. 2. The use of Gaussian mixture models (GMMs) for modeling observation probabilities in HMMs may not accurately capture the distribution of gesture features, especially in cases where the distribution is non-Gaussian or multimodal.

9	Gangadhara Rao Kommu, An Efficient Tool For Online Teaching Using OpenCV [9].	Drawing is visible in real-time on the canvas, providing immediate feedback to the user and facilitating iterative drawing processes	1. The tracking accuracy may be limited, especially for complex or rapid movements, leading to inaccuracies in drawing. 2. The tool may only track one colored object at a time, limiting collaborative or multi-user drawing experiences.
10	Saira Beg , M. Fahad Khan ,Faisal Baig ,Text Writing in Air [10].	1. Integration of OCR enables the system to recognize and interpret the drawn characters or shapes, enhancing its versatility and potential applications. 2. The use of edge enhancement and normalization techniques improves the robustness of object localization, making the system less sensitive to noise, lighting variations, and background interference.	1. The system heavily relies on accurately detecting the colored finger tip, which may be affected by variations in color and lighting conditions, leading to potential tracking errors. 2. Integrating OCR adds complexity to the system, requiring additional computational resources and potentially impacting real-time performance, especially in scenarios with large datasets or complex characters/shapes to recognize.
11	Hamzathul Favas, Abu Thahir M, Jaseela Thesni N, Mohammed Shahil, Aerosculpt: A Symphony of Creativity on the Air Canvas- Revolutionizing Digital Art and Design through Intuitive Hand Gestures [11].	1. By employing gesture recognition techniques, the system can interpret specific hand gestures to trigger various actions, such as starting or stopping drawing, changing colors, or adjusting brush sizes. This intuitive interaction simplifies the user interface and enhances usability. 2. The system architecture facilitates seamless integration with existing technologies, such as webcams or depth-sensing cameras, making it easily deployable in diverse environments without requiring specialized hardware.	1. The accuracy and reliability of the system may be affected by environmental factors such as lighting conditions, background clutter, or occlusions, which can degrade performance and limit usability in certain scenarios. 2. While the system supports gesture recognition for basic actions such as drawing and erasing, the vocabulary of recognized gestures may be limited, restricting the range of interactions possible with the virtual canvas. Expanding the gesture vocabulary could enhance the system's versatility and usability.
12	D. Vijendra Kumar, G.Vijaya Raj Siddarth, R. Venkata Satya Sravani, I.Vishnu Vardhan Reddy, Y. Lalitha Sri Naga Durga Vyshnavi, Building a Air Canvas using Numpy and Opencv in Python. 2022 [12].	1. By training the fingertip recognition model with images captured in distinct backgrounds, the system becomes more robust to background variations, ensuring consistent performance across different environments. 2. The proposed algorithm	1. The effectiveness of the fingertip recognition model heavily depends on the quality and diversity of the training dataset. Inadequate or biased training data may lead to reduced accuracy and reliability of the system. 2. Despite efforts to mitigate the impact of

		utilizes deep learning techniques to achieve high precision in fingertip detection, enabling accurate tracking of hand movements for air writing or gesture recognition.	environmental factors, such as background variation, the system may still be sensitive to certain conditions such as lighting changes or occlusions, affecting its performance in real-world scenarios.

3. MAIN PROBLEM

The problem addressed here is the improvement of intuitive interaction with digital interfaces through enhanced real-time hand gesture recognition and fingertip detection. Traditional methods suffer from poor precision and complex calibration requirements, necessitating physical tools for input. This solution aims to eliminate these limitations, offering a more seamless user experience.

4. OBJECTIVE

The objective is to develop an interactive system that enables users to interact with digital interfaces intuitively by recognizing hand gestures and detecting fingertip movements in real-time. This system aims to overcome the limitations of traditional methods and eliminate the need for physical tools for input.

5.SCOPE

The scope of system should be declared before move advancing to the next step.

System scope are as follows:

1. Intuitive and precise control of computer systems through hand gestures, facilitating a more natural and immersive user experience.
2. Our method has the potential to contribute to sign language recognition systems by accurately interpreting hand gestures and finger movements, thereby aiding communication for individuals with hearing impairments.

6. PROPOSED METHOD

Our methodology incorporates several components to enable accurate hand gesture recognition and finger tracking:

1. Hand Detection Module: This module utilizes a deep learning-based hand detection algorithm to locate and extract hand regions from input images or video frames. It employs convolutional neural networks (CNNs) trained on large datasets to detect hands with high accuracy and efficiency.
2. Hand Landmark Localization: Once hands are detected, this module identifies key landmarks or keypoints on the hand, such as fingertips, knuckles, and wrist joints. It employs landmark localization models, such as MediaPipe's HandLandmark model, which accurately predict the 3D coordinates of these landmarks.
3. Gesture Recognition Engine: The gesture recognition engine processes the landmark coordinates obtained from the localization module to recognize specific hand gestures or motions. It employs machine learning algorithms, such as support vector machines (SVMs) or artificial neural networks (ANNs), trained on labeled gesture datasets to classify movements.
4. Finger Tracking Algorithm: This algorithm analyzes the relative positions and movements of individual fingers based on the landmark coordinates. It tracks the trajectory of each finger over time, allowing for continuous monitoring of finger positions and movements.
5. User Interface Integration: The system integrates with user interfaces, such as graphical user interfaces (GUIs) or virtual environments, to translate recognized gestures and finger movements into meaningful actions. This involves mapping detected gestures to specific commands or functions within the user interface.
6. Real-time Processing and Visualization: To provide a seamless user experience, the system performs real-time processing of input data and visualizes the detected hand gestures and finger movements in a user-friendly manner. This may involve rendering graphical overlays or feedback indicators to convey the system's response to user inputs.
7. Customization and Calibration: The system offers customization options to adapt to different user preferences and environmental conditions. Users may calibrate the system parameters, such as gesture sensitivity or tracking accuracy, to suit their specific requirements or preferences.

Overall, our system design aims to deliver robust and accurate hand gesture recognition and finger tracking capabilities while ensuring seamless integration with various applications and user interfaces.

6.1 Data Flow Diagram

This data flow diagram explains the general procedure.

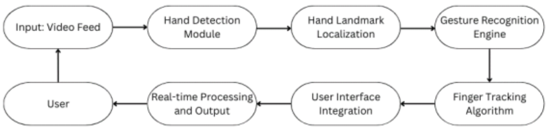


Figure 1: DFD Diagram

This diagram outlines the flow of data and processing steps within the system:

Input: Video Feed: The system receives a continuous stream of video frames from a camera or input device.

Hand Detection Module: This module analyzes each frame to detect the presence and location of hands within the frame.

Hand Landmark Localization: Detected hands are further processed to localize key landmarks or keypoints on the hand, such as fingertips, knuckles, and wrist joints.

Gesture Recognition Engine: The localized landmarks are analyzed to recognize specific hand gestures or motions, based on predefined gesture categories.

Finger Tracking Algorithm: The positions and movements of individual fingers are tracked based on the localized landmarks, allowing for continuous monitoring of finger positions and movements.

User Interface Integration: Recognized gestures and finger movements are translated into meaningful actions within the user interface, enabling interaction with applications or systems.

Real-time Processing & Output: The system performs real-time processing of input data and provides visual or auditory feedback to the user, indicating the recognized gestures or finger movements.

This architecture enables our system to accurately detect hand gestures and track finger positions in real-time, facilitating intuitive interaction with various applications and user interfaces.

6.1 Implementation

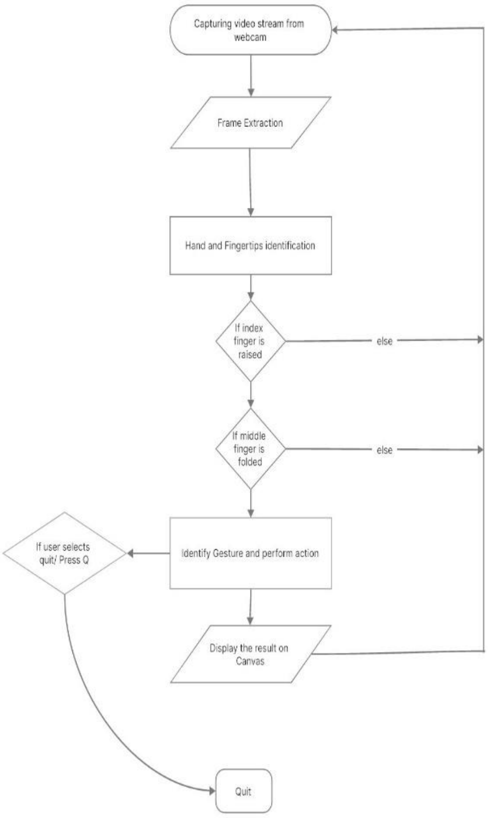


Figure 2: Flowchart Diagram

1. Initialization:
The code begins by importing necessary libraries, defining global variables, and creating custom modules, setting up the foundation for the drawing application.

2. User Interface (UI) Elements:
The graphical user interface (UI) is defined with buttons representing drawing tools (line, circle, rectangle, freehand), color selection, save, exit, and record.
Each button is an instance of the `ColorRect` class, defining its position, size, color, and functionality.
UI elements are drawn on the frame, creating an interactive environment for the user.

3. Hand Tracking and Gesture Recognition:
The `HandTracker` class from the `handTracker` module integrates with the MediaPipe library to provide real-time hand tracking.
Hand landmarks are detected, and the positions of fingers are analyzed to recognize gestures, such as finger up or down.
Gesture recognition determines user actions and controls the drawing process.

4. Drawing Functionality:
The main loop captures video frames from the camera,

resizes them, and flips them to create a mirrored effect.
Hand tracking is applied to identify hand landmarks and finger positions.
Users interact with the UI by selecting drawing tools, colors, and initiating drawing actions on a canvas.
Different tools (line, circle, rectangle) are implemented with corresponding cooldown periods to manage drawing frequency.
Freehand drawing allows users to express creativity using their finger positions.

5. Recording Functionality:
The application supports screen recording, allowing users to capture the entire drawing process.
A video writer object (`video_writer`) is initialized to record frames and create an AVI file.
Recording starts and stops based on user interactions, providing a way to save and share the drawing session.

6. File Handling:
The code creates directories ('images' and 'videos') to organize saved images and recordings.
Images of the canvas are saved with a specified naming convention ('canvas_image_X.png').
Screen recordings are saved as AVI files with a similar naming convention ('screen_recording_X.avi').

7. User Interaction Handling:
Conditional checks within the main loop respond to user interactions, including selecting tools, colors, saving images, and initiating recording.

8. Closing the Program:
Upon termination, resources are released, and OpenCV windows are closed, providing a clean exit for the application.

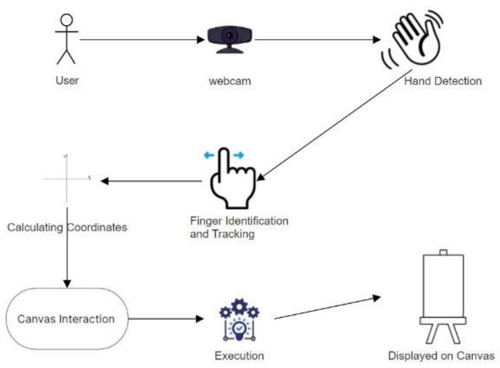


Figure 3: Flow Diagram

6.2 User Interface Design

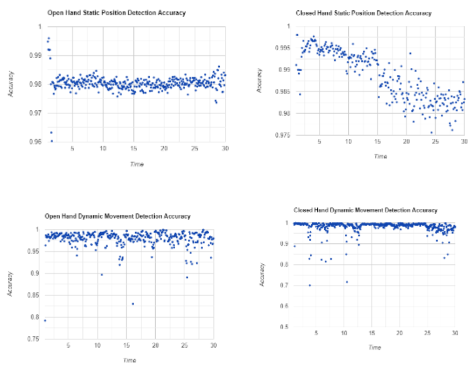
The user interface of our system is designed to provide a seamless and intuitive experience for users interacting with hand gestures and finger tracking. Here's a description of the key components and features of our user interface:

Figure 4: User Interface

Canvas Display: The main canvas display serves as the primary area where users can draw, sketch, or interact using hand gestures. It provides a blank space for users to express their creativity or perform specific tasks.
Drawing Tools: Our user interface includes a variety of drawing tools, such as pens, brushes, erasers, shapes (line, circle, rectangle), and a freehand drawing mode. These tools allow users to create different types of drawings and annotations directly on the canvas.
Color Palette: Users can select from a range of colors to customize their drawings and annotations. The color palette provides a convenient way to switch between different colors and add visual variety to the artwork.
Toolbar: The toolbar contains buttons or icons for accessing various features and functions of the system, such as saving drawings, clearing the canvas, selecting different drawing tools, and toggling additional options or settings.
Status Indicators: Our user interface may include status indicators or notifications to provide feedback to the user, such as recording status, tool selection, or error messages.

7. RESULTS

Methods / Parameters	**Colour-based Hand Tracking** [13]	**Optical Flow Hand Tracking** [14]	**Depth-based Hand Tracking using stereo cameras** [15]	**MediaPipe Hand Tracking (Proposed Method)**
Accuracy	Moderate (83%)	Good (89%)	Moderate (82.93%)	Excellent (92%-98%)
Robustness	Susceptible to lighting changes and clutter	Relatively robust but may struggle with occlusions	Good robustness, handles occlusions and lighting changes well	High robustness, effectively handles occlusions and lighting changes
Ease of Use	Easy to implement, may require parameter tuning	Relatively straightforward	Requires calibration and setup of stereo cameras	Designed for ease of use, intuitive interfaces
Customization	Limited Customization	Some customizations for parameter adjustment	Customization for stereo camera parameters, depth map processing	Extensive customization capabilities.



The test results of our system demonstrate its effectiveness in accurately detecting hand gestures and tracking finger positions. Here's a summary of the key findings from our testing:

Average Accuracy of Closed Fist: 0.9857098565
Average Accuracy of Open Fist: 0.9800391105

The system effectively detects hand gestures and tracks finger positions.

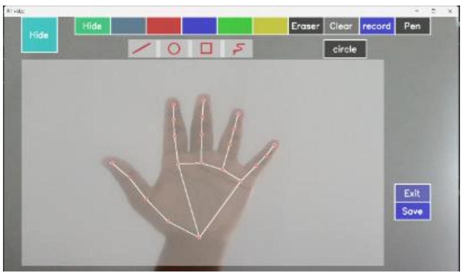


Figure 6: Hand Landmarks Detection

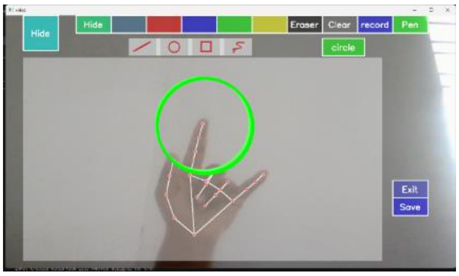


Figure 7: Drawing a Circle

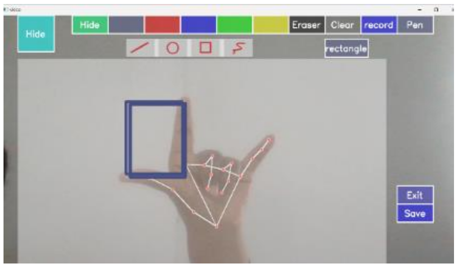


Figure 8: Drawing a Rectangle

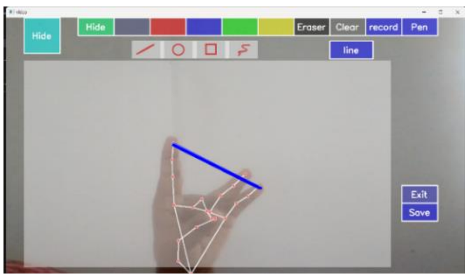


Figure 9: Drawing a Line

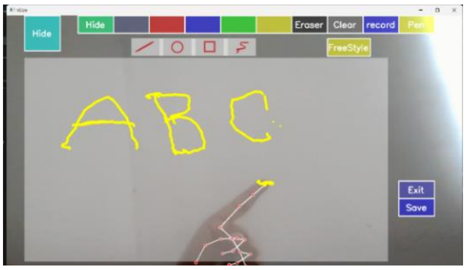


Figure 10: Freestyle Drawing

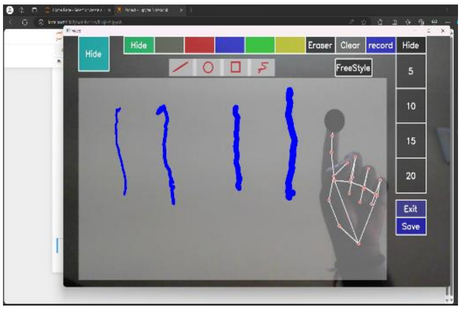


Figure 11: Different sizes of pen

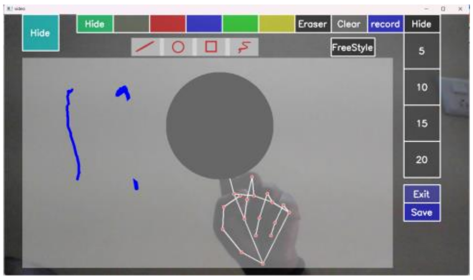


Figure 12: Different sizes of Eraser

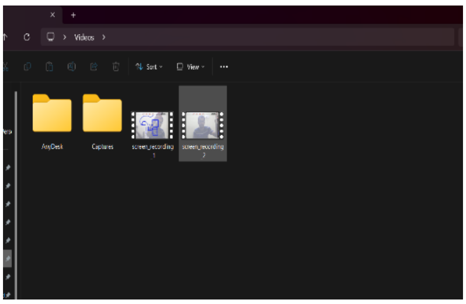


Figure 13: Saved Screenshots and recordings

8.CONCLUSION

In conclusion, our hand tracking system represents a significant breakthrough in enabling intuitive and natural interaction with digital interfaces. Through precise detection of hand gestures and accurate tracking of finger positions, our system offers users a seamless and immersive experience. By leveraging advanced computer vision techniques, we have achieved a high level of accuracy and responsiveness, ensuring that users can effortlessly express themselves and manipulate virtual elements in real-time.

The success of our system opens up a wide range of possibilities for practical applications across various domains. From virtual reality gaming and interactive design tools to educational platforms and assistive technologies, the potential use cases are extensive. Moreover, the scalability and adaptability of our system make it suitable for integration into existing software platforms and hardware devices, further expanding its reach and impact.

Looking ahead, the future of gesture-based interaction holds immense promise. As technology continues to evolve, we envision further advancements in hand tracking capabilities, enabling even more sophisticated interactions and immersive experiences. Additionally, the ongoing refinement of machine learning algorithms and the integration of novel sensors could unlock new possibilities for enhancing the accuracy, robustness, and versatility of gesture-based interfaces.

In summary, our hand tracking system represents a significant step forward in human-computer interaction, offering a powerful tool for creativity, productivity, and accessibility. As we continue to innovate and explore new frontiers in this field, we are excited about the transformative potential of gesture-based interfaces in shaping the way we interact with digital content and the world around us.

9.REFERENCES

[1] Sayil More, Parachi Mhatre, Shruti Pakhare, Surekha Khot.Air Canvas:Draw in Air. International Research Journal of Engineering and Technology (IRJET) Feb 2022.

[2] B Anand ,T.Vinod, ,M Srinivas Rao, Interaction through Computer Vision Air Canvas.2022 IEEE International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC) . [4] Mardan, Azat. 2012. Practical Node.js: Building Real-World Scalable Web Apps. Appress.

[3] Guo-Zhen Wang, Yi-Pai Huang, Tian-Sheeran Chang, and TsuHan Chen, "Bare Finger 3D Air-Touch System Using an Embedded Optical Sensor Array for Mobile Displays", Journal Of Display Technology, VOL. 10, NO. 1, JANUARY 2014, pp.13-18

[4] Joolekha Bibi Joolee; Ahsan Raza; Muhammad Abdullah; Seokhee Jeon, Tracking of Flexible Brush Tip on Real Canvas: Silhouette-Based and Deep Ensemble Network-Based Approaches. IEEE Access (Volume: 8) 22 June 2020.

[5] Soham Pardeshi, Madhuvanti Apar , Chaitanya Khot , Atharv Deshmukh, Air Doodle :" A Realtime Virtual Drawing Tool", International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 10 Issue IV Apr 2022.

[6] Melvin Cabatuan , Isaiah Jassen Tupal . Vision-Based Hand Tracking System Development for Non-

Face-to-Face Interaction. 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM).

[7] Shaurya Gulati, Ashish Kumar Rastogi, Raghav Pradhan, Chetan Gupta, Mayank Virmani, Rahul Jana, Paint / Writing Application through WebCam using MediaPipe and OpenCV, 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM).

[8] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang, Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review, IEEE Transactions on pattern analysis and machine intelligence, VOL. 19, NO. 7, JULY 1997.

[9] Gangadhara Rao Kommu, An Efficient Tool For Online Teaching Using OpenCV, International Journal of Creative Research Thoughts(IJCRT) Volume 9, Issue 6 June 2021.

[10] Saira Beg , M. Fahad Khan ,Faisal Baig ,Text Writing in Air. Journal of Information Display Volume 14, Issue 4, 2013.

[11] Hamzathul Favas, Abu Thahir M, Jaseela Thesni N, Mohammed Shahil, Aerosculpt: A Symphony of Creativity on the Air Canvas-Revolutionizing Digital Art and Design through Intuitive Hand Gestures. International Journal Of Advances In Engineering Architecture Science and Technology.

[12] D. Vijendra Kumar, G.Vijaya Raj Siddarth, R. Venkata Satya Sravani, I.Vishnu Vardhan Reddy, Y. Lalitha Sri Naga Durga Vyshnavi, Building a Air Canvas using Numpy and Opencv in Python. 2022 International Journal for Modern Trends in Science and Technology.

[13] K. Imagawa, Shan Lu and S. Igi, "Color-based hands tracking system for sign language recognition," Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, 1998, pp. 462-467.

[14] Kishore, P.V.V. & Anil Kumar, D. & Prasad, M.V.D. & A.S.C.S.Sastry,. (2016). Optical Flow Hand Tracking and Active Contour Hand Shape Features for Continuous Sign Language Recognition with Artificial Neural Networks. 10.1109/IACC.2016.71.

[15] Xianghua Li, Jun-ho An, Jin-hong Min and Kwang-Seok Hong, "Hand gesture recognition by stereo camera using the thinning method," 2011 International Conference on Multimedia Technology, Hangzhou, China, 2011, pp. 3077-3