

Air-Canvas: Real-Time Gesture Recognition System for Interactive Digital Writing

**Kiran Yerolkar*1, Nirav Otageri*2 , Meghana Pawar*3 , Aditi Kale*4
Guide –Prof. Dipika Mam*5, HOD Prof.Nita Pawar*6**

*1,2,3,4Student, Ajeenkya D. Y. Patil School Of Engineering, Computer Engineering Diploma, Charholi, Pune, India.

*5,6Professor Ajeenkya D. Y. Patil School Of Engineering, Computer Engineering Diploma, Charholi, Pune, India.

Abstract

With the rapid advancement of technology, every sector is moving toward modernization and automation. The development of smart devices and intelligent systems has enabled virtual control through natural human gestures. While using traditional paint applications, users often find it difficult to express their creativity freely using a mouse or touchpad. Many people imagine drawing simply by waving their hands in the air.

The Air Canvas project focuses on developing a gesture-based virtual drawing system that converts hand movements into digital strokes in real time. This project is built using the Open Computer Vision Library (opencv) and mediapipe, which together enable accurate hand tracking and gesture recognition. The user's hand acts as a virtual pen, allowing them to draw lines, rectangles, circles, and freehand sketches. Additionally, specific gestures are used to activate tools such as eraser, color selection, image insertion, and PDF annotation.

Unlike existing systems that require complex hardware setups or time-consuming processing techniques, this project uses an optimized and lightweight methodology. The system camera captures live video input, detects hand landmarks using mediapipe, and translates finger movements into drawing commands efficiently.

This project demonstrates the practical implementation of computer vision in human-computer interaction (HCI). It reduces dependency on physical input devices and provides a more intuitive and interactive user experience. The Air Canvas system can be extended for educational purposes, digital presentations, virtual classrooms, and touchless interaction systems in modern smart environments.

Keywords: Air Canvas, Hand gesture recognition , Computer vision , Mediapipe, Opendv, Human-computer interaction (hci), Virtual drawing system, Real-time hand tracking, Gesture-based control, Touchless interaction, Digital annotation, Pdf annotation, Motion-to-digital conversion, Smart interactive systems

Date of Submission: 12-02-2026

Date of Acceptance: 22-02-2026

I. Introduction

Traditionally, digital painting and drawing were performed using input devices such as a mouse or touchpad, which can be stressful and less intuitive for users. Although touch screen laptops provide a more natural drawing experience, they are relatively expensive and not accessible to everyone. To overcome these limitations, hand tracking — specifically finger tracking — can be used as a virtual input tool that functions similarly to a keyboard or mouse. This technology is widely applied in fields such as Virtual Reality, augmented reality, and sign language recognition.

The Air Canvas system is a hands-free digital drawing platform that utilizes a system camera along with OpenCV and MediaPipe to recognize and map hand gestures in real time. The user's finger acts as a virtual brush or pen to draw on the canvas or annotate PDF documents. The brush size can be adjusted, and pen colors can be changed by hovering the pointer over built-in virtual buttons displayed on the screen.

Computer vision techniques are employed to detect hand landmarks and enable drawing of various shapes such as lines, rectangles, and circles. The system is developed using the Python programming language. The camera captures live input, MediaPipe tracks finger positions, and OpenCV's built-in functions are used to render shapes on the digital canvas.

Additionally, the system allows users to:

- ✓ Open and annotate PDF documents
- ✓ Underline or highlight important content
- ✓ Insert images into the canvas
- ✓ Save the canvas work as an image file

- ✓ This project demonstrates the integration of gesture recognition and computer vision to create a touchless, interactive human-computer interface.

- **System Requirements**

1. Hardware Requirements

- ✓ The minimum hardware requirements to execute the system are:
- ✓ Processor – Intel i5 or equivalent
- ✓ RAM – 4 GB (Minimum)
- ✓ Storage – 1 GB free space
- ✓ Web Camera

2. Software Requirements

- ✓ Operating System – Windows 10 or above
- ✓ Programming Language – Python
- ✓ Libraries/Frameworks – OpenCV, MediaPipe, NumPy, PyMuPDF
- ✓ Front End – Python Tkinter (for file dialog) and OpenCV window interface

3. Functional Requirements

- ✓ The system must be able to:
- ✓ Capture live input using the system camera
- ✓ Detect hand positions and fingertip coordinates
- ✓ Allow selection of different shapes, colors, and brush sizes
- ✓ Draw shapes on the digital canvas
- ✓ Erase drawings using gesture-based control
- ✓ Save the canvas work as an image file
- ✓ Open PDF documents and allow annotation/editing

4. Non-Functional Requirements

- ✓ Reliability: The system should consistently detect hand gestures with minimal errors.
- ✓ Scalability: The system should support additional features such as gesture-based navigation or multi-user interaction in the future.
- ✓ Maintainability: The code structure should allow easy modification and feature updates.
- ✓ Usability: The interface should be simple, intuitive, and user-friendly.
- ✓ Availability: The system should function whenever the required hardware and software environment is available.

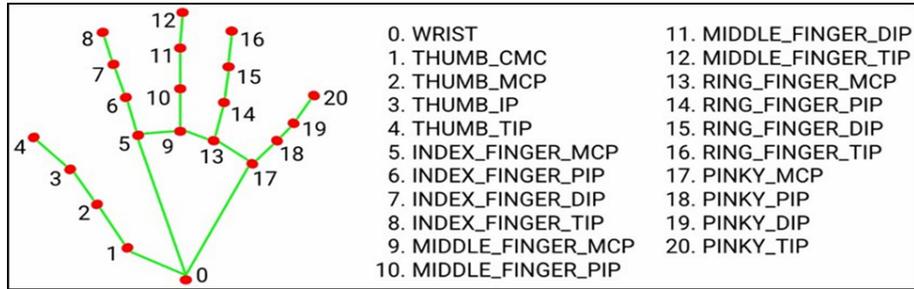


Fig -2: Hand landmarks

II. Problem Statement

The increasing reliance on digital and remote learning environments highlights the limitations of traditional educational tools, such as whiteboards and projectors, which require physical interaction and hinder both teachers' and students' flexibility and creativity. As distance learning becomes more prevalent, there is a pressing need for interactive and engaging solutions that can effectively support varied learning styles while addressing the challenges of non-interactive formats.

III. Objective

- Gesture Recognition:** Implement AI-based hand gesture recognition to allow users to write and draw in the air.
- Real-Time Interaction:** Ensure that the system provides instant feedback and updates on the virtual board as gestures are performed.
- Advanced Features:** Integrate advanced functionalities such as multi user collaboration, voice commands, 3D drawing capabilities, and cloud synchronization.
- User Experience:** Design an intuitive user interface that mimics traditional whiteboard interactions while incorporating innovative features for enhanced user engagement.

IV. Literature Review

Sr. No.	Name of the document	Technology Used	Limitation
1	Saurabh uday Saoji , AIR CANVAS APPLICATION USING OPENCV AND NUMPY IN PYTHON (2021), International Research Journal of Engineering and Technology (IRJET) Volume: 08 Issue: 08 Aug 2021	Python Libraries: OpenCV,NumPy Deep Learning Models: Single Shot Detector (SSD), Faster R-CNN	Gesture Control Challenges, Air Character Dataset, Background Interference, Gesture Control Challenges
2	Shreyas Amol Sandbhor, Himanshu Shekatkar, Aniket Nawalkar, Ms. Sucheta Navale, Sandbhor, Shreyas and Shekatkar, Himanshu and Nawalkar, Aniket and Navale, Sucheta, Survey Paper on Air canvas Using OpenCV (February 5, 2024).	Python Libraries: OpenCV,NumPy, Tkinter Framework: MediaPipe	Background Interference, Lack of Advanced Features, Hardware Dependency
4	Tamalampudi Hema Chandhan, Kavin Kumar R, Nalin Raj, Neelam Nanda Kishore Reddy, Dr.Mohammed Zabeeulla Air Canvas: Hand Tracking Using OpenCV and MediaPipe, 1st - International Conference on Recent Innovations in Computing, Science & Technology	Python Libraries: OpenCV,NumPy, Tkinter, Framework: MediaPipe	Required Lighting Conditions, Hardware Dependency, Occlusions
5	Mitesh Ikar, Gayatri Jagnade, Nikita Chaudhari, Computer Vision-based Air Canva Virtual Paint, Conference: International Journal of Trend in Research and Development Volume: Volume 10(2) Issue:April 2023	Python Libraries: OpenCV,NumPy, Framework: MediaPipe	Limited Gesture Range, Hardware Dependency, Lighting and Background Sensitivity

V. Methodology

Existing air canvas applications often rely on training large datasets of hand images to detect finger positions and recognize gestures. This methodology requires significant data storage and computational resources. Moreover, it may result in incorrect predictions due to variations in background conditions, lighting, and differences in skin tone. Some existing systems also depend on traditional image processing techniques such as threshold-based segmentation and database-driven image comparison, which can reduce accuracy and robustness in real-time applications.

- VI. To overcome these limitations, the proposed system utilizes the MediaPipe framework for efficient and reliable hand tracking. In the proposed approach, MediaPipe detects hand landmarks using a pre-trained machine learning model and extracts precise finger coordinates directly from live video input. Instead of relying on large stored datasets or threshold-based methods, the system processes real-time landmark positions to determine finger movements and gestures.
- VII. This approach significantly improves accuracy, reduces computational complexity, and enhances performance under varying lighting and background conditions. The workflow of the proposed system includes capturing live video through the system camera, detecting hand landmarks, identifying fingertip positions, mapping gestures to drawing actions, and rendering shapes on the digital canvas.

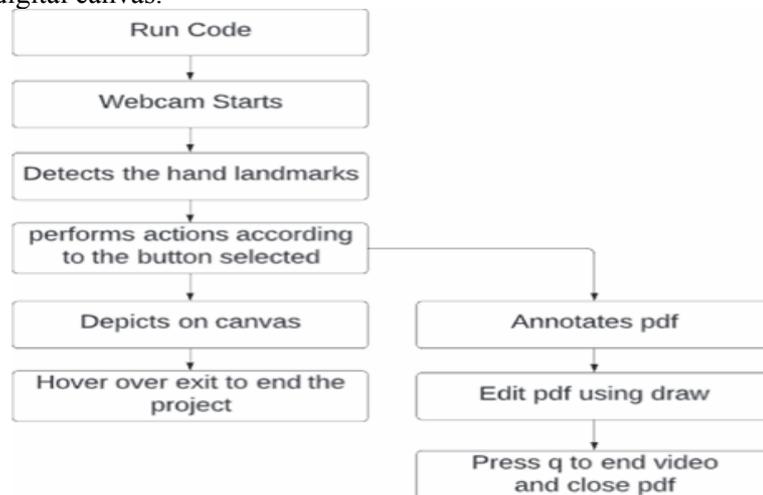


Figure 1: Flowchart of Project.

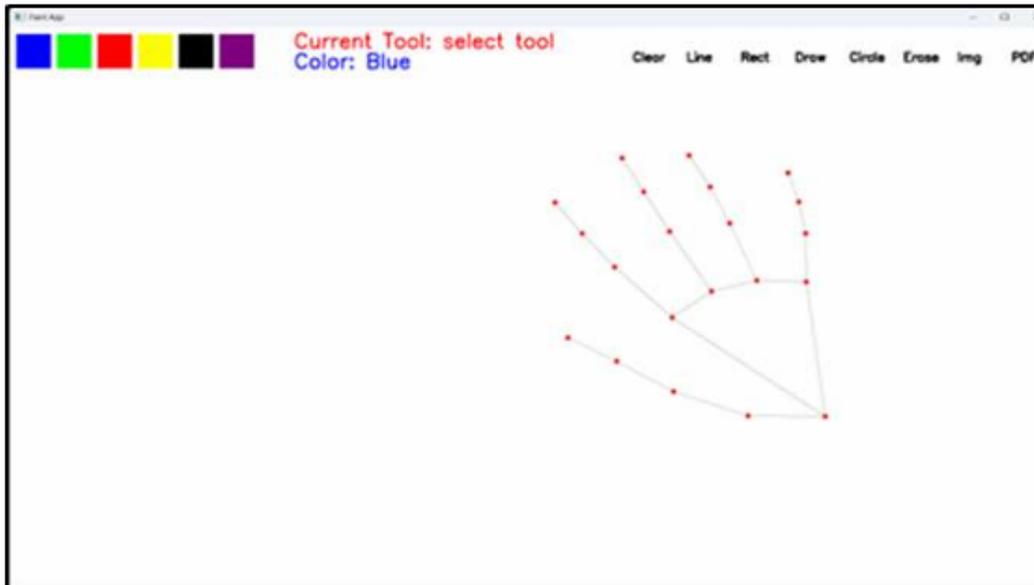
VI. System Implementation

The Air Canvas system is implemented using Python programming language with the integration of OpenCV and MediaPipe libraries. The implementation focuses on real-time hand tracking, gesture recognition, and digital drawing operations. The system architecture is designed to ensure smooth performance, accurate gesture detection, and efficient rendering of graphical elements.

1. Hand Detection and Tracking

The system uses MediaPipe’s hand tracking module to detect and track hand landmarks from live video input. The webcam continuously captures frames, which are processed in real time. MediaPipe identifies 21 hand landmarks for each detected hand, including fingertip coordinates. These landmarks are used to determine finger positions and recognize gestures.

The fingertip of the index finger is treated as the primary drawing pointer. Based on the relative positions of other fingers, the system determines whether the user intends to draw, select a tool, erase content, or change color.



2. Gesture Recognition Mechanism

Gesture recognition is implemented by analyzing the spatial relationship between detected landmarks. For example:

- If the index finger is raised, drawing mode is activated.
- If multiple fingers are detected in a specific configuration, tool selection is triggered.
- Specific regions on the screen act as virtual buttons for selecting shapes, colors, and utilities.

This approach eliminates the need for threshold-based image segmentation and reduces sensitivity to lighting and background variations.

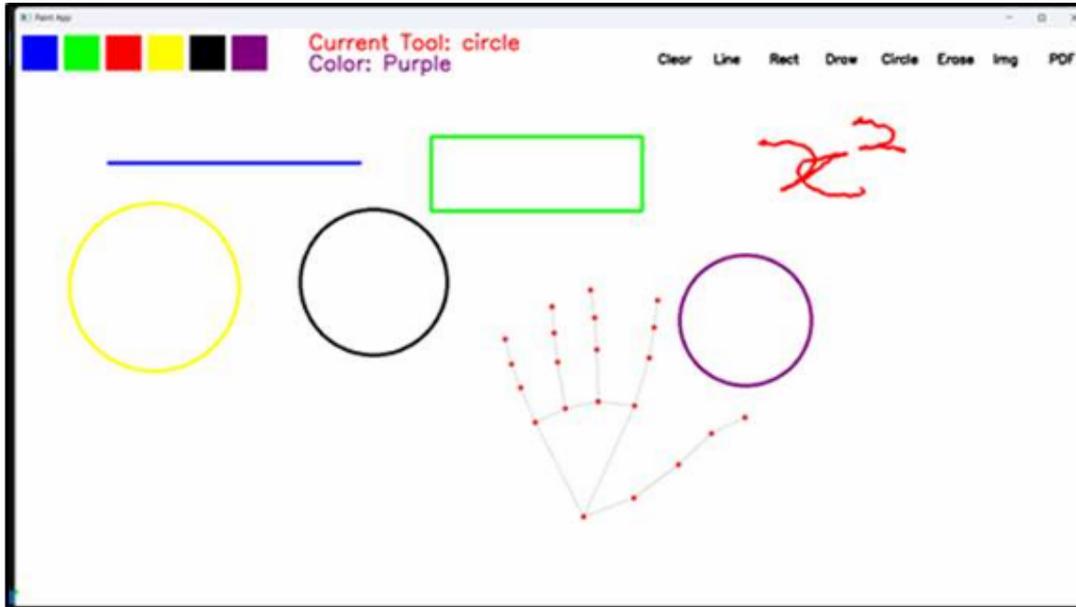
3. Digital Canvas and Drawing Module

A virtual canvas is created using NumPy arrays. OpenCV drawing functions such as `line()`, `rectangle()`, and `circle()` are used to render shapes. The drawing operations are mapped to fingertip coordinates detected by MediaPipe.

The system supports:

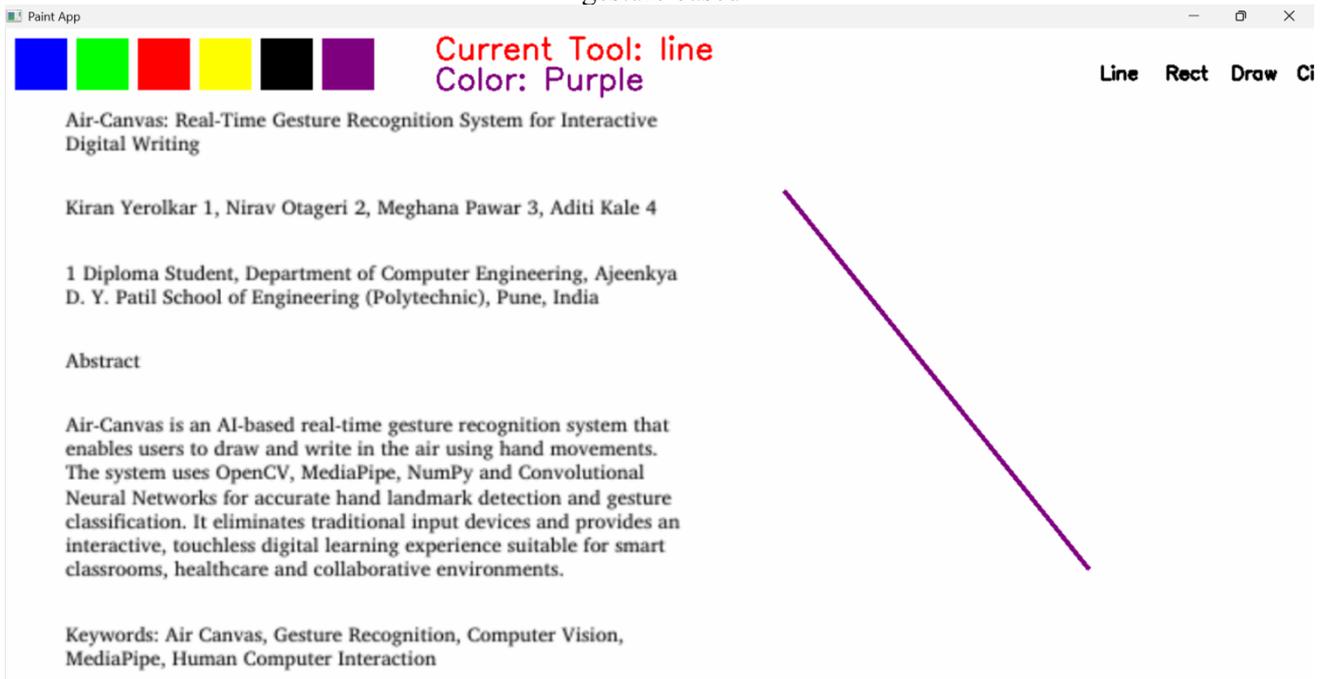
- Freehand drawing
- Line drawing
- Rectangle and circle drawing
- Erasing functionality
- Color selection
- Adjustable brush thickness

The drawing is displayed in real time by overlaying the canvas onto the live video feed.



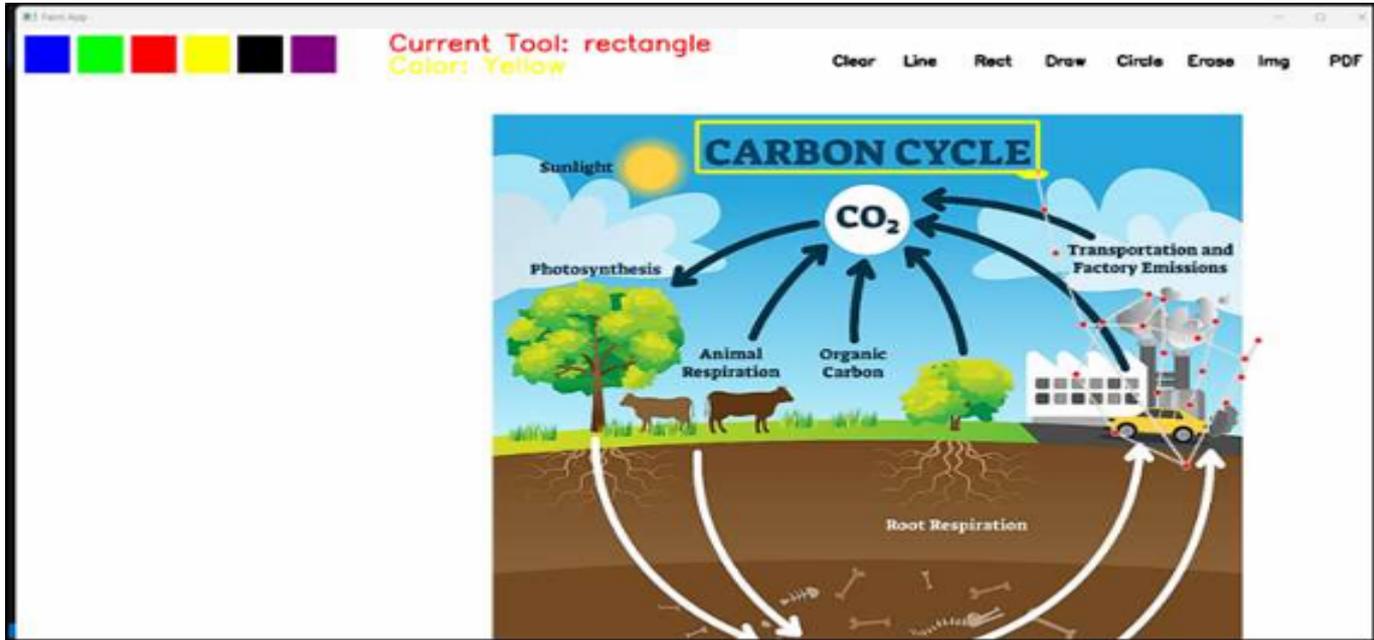
4. PDF Annotation Module

The system integrates PyMuPDF to enable PDF handling. When a user selects a PDF file, the pages are converted into image format and displayed on the canvas. Users can annotate, underline, or mark content directly using gesture-based drawing. Page navigation is implemented through scroll events or gesture based



5. Image Insertion Module

Users can insert images into the canvas using a file selection dialog. The system ensures only supported image formats are selected. The inserted image is resized proportionally to fit within the canvas dimensions.



6. Saving the Canvas Output

The system allows users to save the final canvas output as an image file. This feature ensures that drawings and annotations can be preserved for later use.

7. Performance Optimization

To ensure smooth execution, the system:

- Uses real-time frame processing
- Minimizes computational overhead
- Maintains efficient memory management
- Operates effectively under varying lighting conditions

The implementation successfully integrates computer vision and gesture recognition techniques to create a responsive, touchless interactive drawing system.

VII. Functional Description:

System Initialization:

Required Equipment:

Camera/Sensor will be tracking hand movements. Loading libraries and dependencies such as OpenCV, Media Pipe. Drawing Configuration by specifying the size and resolution of a canvas, along with any additional parameters.

Capture Video Frame

Begin the video stream, and the program will start grabbing frames from the camera. Each frame can be preprocessed in advance, depending on your requirements for accuracy. Convert to a format suitable for further analysis and processing.

Implement Hand Tracking:

Use Media Pipe's hand tracking model to detect and track the key hand landmarks. Collect the landmarks such as fingers, palm, and wrist from the model's output. Apply transformations, if appropriate, to

change the coordinates of the detected hand landmarks.

Translate Hand Movement onto Canvas:

Establish a connection between hand movements and the coordinates of the canvas. Calculate the position of the hand on the canvas using detected landmarks. The position of a virtual drawing tool-a virtual pen or brush-is updated based on user's hand movements.

Drawing on the Canvas:

Implement the functionality of drawing by, for example a line or any shape. Note that this is a drawing request, for instance by detecting a click button or a specific gesture. Update the canvas in real time with the movement of the hand, thus allowing air drawing.

Presentation of the Canvas:

Render the canvas on the display continuously as any drawn content appears. In addition, ensure smooth updates of the canvas in response to hand movement so that visual feedback is provided in real-time for noninterrupted drawing.

Handling user inputs:

Implement the functionality of some user interactions like canvas delete or switching between the tools. Identify hand gestures or commands by which the system should respond for the action to be performed. Monitor the input from the

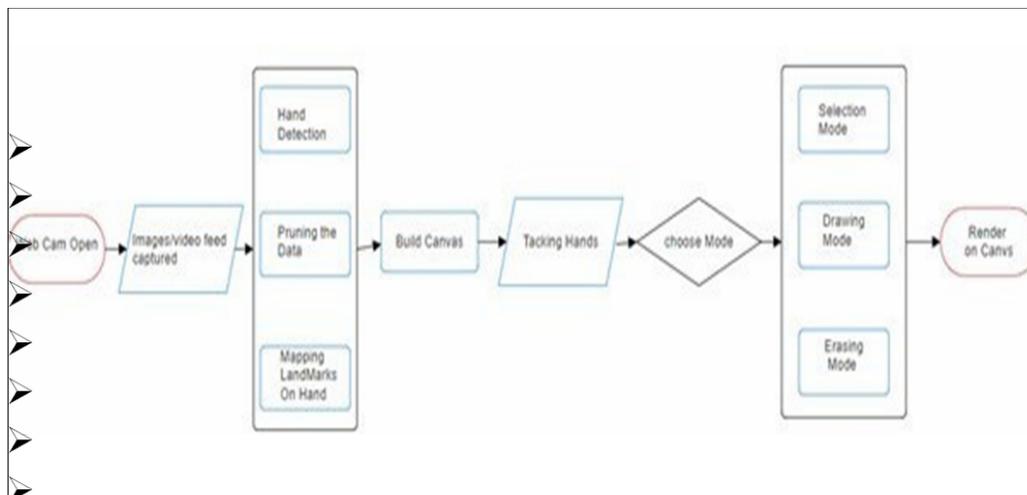


Fig. Work Flow Diagram

VIII. Future Scope

- **Gesture Recognition:** AI-driven hand gesture tracking using standard cameras or sensors.
- **Virtual Canvas:** Real-time rendering of user input, allowing for drawing, writing, erasing, and using various tools.
- **User Interface:** Design and development of an intuitive UI for users to interact with the canvas and select drawing tools.
- **Collaboration:** Multi-user functionality for real-time collaboration on a shared virtual canvas.
- **Advanced Features:** Voice commands for tool selection and actions, cloud- based session saving, and 3D drawing capabilities.
- **Cross-Industry Applications:** Usable in education, healthcare, design, and VR environments.
- **Accessibility:** Special features for users with physical disabilities, such as gesture-only or voice-controlled interaction.

IX. Reference List / Bibliography

- [1]. Saurabh Uday Saoji , Air Canvas Application Using Opencv And Numpy In Python (2021), International Research Journal Of Engineering And Technology (Ijret) Volume: 08 Issue: 08 | Aug 2021 Available At (Pdf) Air Canvas Application Using Opencv And Numpy In Python (Researchgate.Net)
- [2]. Tamalampudi Hema Chandhan, Kavin Kumar R, Nalin Raj, Neelam Nanda Kishore Reddy, Dr.Mohammed Zabeeulla Air Canvas: Hand Tracking Using Opencv And Mediapipe, 1st International Conference On Recent Innovations In Computing, Science & Technology Isbn:978-81-963209-0-4, Available At Air Canvas: Hand Tracking Using Opencv And Mediapipe By Tamalampudi Hema Chandhan, Kavin Kumar R, Nalin Raj, Neelam Nanda Kishore Reddy, Mohammed Zabeeulla A N :: Ssrn
- [3]. Shreyas Amol Sandbhor, Himanshu Shekatkar, Aniket Nawalkar, Ms. Sucheta Navale, Sandbhor, Shreyas And Shekatkar, Himanshu And Nawalkar, Aniket And Navale, Sucheta, Survey Paper On Aircanvas Using Opencv (February 5, 2024). Available At Ssrn: <https://ssrn.com/abstract=4716593> Or <http://dx.doi.org/10.2139/ssrn.4716593>
- [4]. Mitesh Ikar, Gayatri Jagnade, Nikita Chaudhari, Computer Vision-Based Air Canva Virtual Paint, Conference: International Journal Of Trend In Research And Development Volume: Volume 10(2) Issue:April 2023 Available At (Pdf) Computer Vision-Based Air Canva Virtual Paint (Researchgate.Net)
- [5]. B.Venugopal, Ch.Seshagirirao, B.Ajaykumar. D.Prakash, Md.Shakeelahmed, Air Canvas Application, International Research Journal Of Engineering And Technology (Ijret) Volume: 10 Issue: 11 | Nov 2023, Available At Air Canvas Application | Pdf (Slideshare.Net)