ISSN: 2454-9940

Air Canvas
A Real-Time Color-Based Drawing Interface

Mr. RAGHAVENDRA M I
M.Tech (Ph.D) Assistant Professor

Department of CSE (Data Science)
Sphoorthy Engineering College (JNTUH)
Hyderabad, India.
Email: rmichangi@gmail.com

S. Samhitha

Department of CSE (Data Science)
Sphoorthy Engineering College (JNTUH)
Hyderabad, India.
Email: samreddysamhitha5254@gmail.com

R. Hari Teja

Department of CSE (Data Science)
Sphoorthy Engineering College (JNTUH)
Hyderabad, India.
Email: hariteja.ramasahayam@gmail.com

S. Akshay Prem

Department of CSE (Data Science)
Sphoorthy Engineering College(JNTUH)
Hyderabad, India.
Email: akshayprems123@gmail.com

Dhanush Reddy

Department of CSE (Data Science)
Sphoorthy Engineering College(JNTUH)
Hyderabad, India.
Email: dhanushreddy@gmail.com

ABSTRACT

The Air Canvas project is a real-time, touch-free drawing application that enables users to draw in the air using a red-colored object as a virtual pen. By leveraging computer vision techniques with OpenCV and a webcam, the system detects and tracks the object's position using HSV color segmentation. Through contour detection and centroid calculation, the path of the object is captured and translated into digital strokes, which are rendered both on the live video feed and a separate blackboard canvas. Morphological operations help clean the binary mask, ensuring smooth and accurate tracking. Users can interact with the system via simple keyboard inputs to quit or clear the canvas.

This system showcases a natural and intuitive human-computer interaction approach, eliminating the need for physical input devices. Its potential applications span across digital art, interactive learning, gesture-based interfaces, and assistive technology. Designed with simplicity and accessibility in mind, Air Canvas offers a hygienic, engaging alternative for creative expression and virtual collaboration, particularly relevant in contactless environments and educational settings.

accuracy. Furthermore, it includes a user-friendly interface that allows investors to visualize model performance, compare predictions with actual stock prices, and simulate investment outcomes.

INTRODUCTION

The increasing demand for natural, intuitive, and contactless Human-Computer Interaction (HCI) has highlighted the limitations of traditional input devices like the mouse, keyboard, and touchscreen. These conventional tools often require physical contact, which can pose hygiene concerns, especially in shared environments, and may not be accessible to individuals with certain physical disabilities. To address these challenges, the Air Canvas system offers an innovative, touch-free method for digital interaction. It allows users to draw in the air using a simple colored object, such as a red marker cap, as a virtual pen.

The system works by utilizing a webcam to capture real-time video of the user's hand movements. It leverages OpenCV, a powerful open-source computer vision library, to process each frame of the video. The red-colored object is detected by converting the image from BGR to HSV (Hue, Saturation, Value) color space, which is more effective for isolating specific colors under different lighting conditions. Once the red color is identified, the system uses contour detection to find the shape and location of the object. It then tracks the center of the contour across frames to map the movement. As the object moves through the air, the software interprets its path and draws corresponding lines on a virtual canvas displayed on the screen. This creates the effect of drawing digitally without touching any physical surface. The Air Canvas system not only provides a creative and interactive experience but also opens up numerous practical applications. These include digital art creation, remote teaching and presentations, gesture-controlled user interfaces, and assistive technologies for individuals with mobility challenges. Overall, Air Canvas demonstrates a forward-thinking approach to HCI, combining simplicity, accessibility, and hygiene in a versatile and engaging way.

EXISTING SYSTEM:

Air canvas systems represent a significant advancement in the field of **Human-Computer Interaction (HCI)** by enabling users to interact with digital content in a **contactless and intuitive** manner. These systems allow users to draw or write in mid-air using simple hand gestures or a colored object (acting as a virtual stylus), with their actions digitally reproduced on a screen. The approach offers several benefits, including enhanced hygiene, accessibility for users with disabilities, and a more natural interaction experience compared to traditional input methods like keyboards, mice, or touchscreens.

At the heart of an air canvas system are two core technologies: **gesture recognition** and **stylus tracking**. While gesture recognition interprets specific hand shapes or movements to trigger commands (such as "clear screen" or "change color"), stylus tracking focuses on following the precise motion of a specific object—often a finger tip, marker cap, or another colored item used to draw or write. When combined effectively, these technologies enable a **multi-functional, intelligent, and interactive drawing experience** that goes beyond basic mid-air scribbling.

Our research introduces an enhanced air canvas system that tightly integrates both gesture recognition and stylus tracking using **OpenCV** and **machine learning techniques** in Python. The system uses a standard webcam to capture real-time video frames. These frames are processed using color-based segmentation and converted to the **HSV (Hue, Saturation, Value)** color space to improve robustness under varying lighting conditions. Object detection and **contour analysis** are applied to isolate the stylus (typically identified by its distinctive color), enabling the system to track its **exact position and movement** across the screen. This movement is then translated into **smooth, real-time digital strokes** on a virtual canvas.

In parallel, the system also employs **gesture recognition** techniques. Using frame-by-frame analysis and possibly machine learning classifiers, it detects specific **hand gestures** that perform certain actions—such as starting/stopping drawing, clearing the canvas, or switching between modes (like writing and erasing). These gestures are recognized based on hand shape, motion patterns, or changes in finger configuration, adding an **intelligent control layer** to the system.

By combining gesture-based controls with precise stylus tracking, the system creates a more **interactive and flexible user experience**. For example, a user can start drawing with a simple gesture, draw with high accuracy using the tracked stylus, and then use another gesture to save the artwork or clear the screen—all without touching any physical surface.

ADVANTAGES:

1. Contactless Interaction

- These models are well-established and backed by decades of academic research.

2. Natural and Intutive Interface

- Mimics real-world hand movements, providing a user-friendly experience that feels like drawing or writing with a pen in the air.

3. Improved Accessibility

- Offers an alternative input method for individuals with physical disabilities who may find traditional input devices difficult to use.

4. Real-Time , Accurate Tracking

- Combines object detection and optimized contour analysis to track the stylus position precisely and smoothly in real time.

5.Multi-Functional Gesture Control

- Hand gestures can trigger different system actions (e.g., start/stop drawing, clear canvas, switch modes), adding flexibility and interactivity.

6. Enhanced Creativity and Flexibility

- Users can freely sketch, write, or interact with digital elements, promoting creativity in digital art, remote teaching, and design.

7. No Need for Specialized Hardware

- Works with a standard webcam and colored object, making it cost-effective and widely accessible without requiring expensive devices like digital pens or gloves.

8. Robust Performance Under Various Conditions

- HSV color space and real-time filtering techniques improve system stability across different lighting and background environments.

9. Wide range of application

- Suitable for digital art, remote education, smart classrooms, gesture-based interfaces, and assistive technologies.

10. Hands-Free command Execution

- Gestures allow users to control system functions without interrupting their flow, making the interaction more seamless and efficient.

DISADVANTAGES:

1. Dependence on Lighting Conditions

- Despite using HSV color space, extreme lighting variations or poor illumination can still affect color detection and tracking accuracy.

2. Background Interference

- Complex or cluttered backgrounds may contain colors similar to the stylus, leading to false detections or unstable tracking.

3. Limited Gesture Recognition Accuracy

- Without advanced machine learning models or depth sensing, recognizing complex or subtle gestures may result in false positives or missed commands.

4. Lack of Depth Perception

- A 2D webcam cannot accurately detect movement in the z-axis (depth), limiting interaction to a flat plane and reducing the realism of the in-air drawing experience.

5. User Fatigue

- Continuous hand movement in mid-air for extended periods can cause fatigue, making it impractical for long-duration tasks like detailed artwork or lengthy note-taking.

6. Calibration Requirements

- The system may require fine-tuning (e.g., color thresholds, contour size limits) for each environment or user to function optimally.

7. Limited Precision Compared to Traditional Tools

- The accuracy and smoothness of air-drawn strokes may still lag behind what can be achieved using physical tools like styluses or tablets.

8. Lack of Tactile Feedback

- Users don’t get the physical feedback (e.g., resistance, texture) they would from traditional drawing surfaces, which can affect control and precision.

9. Hardware Dependency

- Relies on a camera with sufficient frame rate and resolution. Low-quality webcams may lead to latency, jittery tracking, or blurred input**.**

10. Limited Environmental Adaptability

- Outdoor or very bright environments can hinder detection due to reflective surfaces and uncontrolled lighting.

PROPOSED SYSTEM:

The proposed **Air Canvas system** is an innovative, contactless drawing interface that enables users to create digital illustrations in the air using a colored object—such as a red-tipped pen—by tracking its motion via a webcam. Designed as a more hygienic and intuitive alternative to traditional input tools like keyboards, mice, or touchscreens, this system offers an accessible solution particularly suited for educational, artistic, or assistive environments. The system operates by capturing real-time video input from a standard webcam, which is then processed frame-by-frame using computer vision techniques implemented with OpenCV in Python. Each video frame is converted from BGR to the HSV (Hue, Saturation, Value) color space, which is more robust for color detection under varying lighting conditions.

To detect the red-colored stylus, two HSV ranges are defined to account for the hue values of red that wrap around the HSV spectrum. A binary mask is generated for each range and combined to isolate red-colored regions in the image.

This mask is then refined using morphological operations such as erosion, opening, and dilation to reduce noise and enhance the visibility of the drawing tool. Contour detection is then applied, with the assumption that the largest contour represents the user's stylus. The system calculates the centroid of this contour and continuously tracks it. To render drawing strokes, a double-ended queue (deque) is used to store the history of these centroid positions, which are connected as lines to simulate smooth and continuous drawing. These strokes are displayed in real time over the camera feed for user feedback and simultaneously on a virtual canvas for persistent output.

The user can interact with the system using keyboard inputs—for instance, pressing 'c' clears the canvas, while 'q' exits the program. Despite its simplicity, the system lays a solid foundation for more advanced gesture-based drawing interfaces. It opens possibilities for further development, including multi-color tracking, gesture-based command recognition, pressure sensitivity simulation, and integration with augmented or virtual reality (AR/VR) environments. The Air Canvas is not only a demonstration of the potential of computer vision in human-computer interaction but also a practical tool for creative, educational, and assistive applications where touchless control is preferred or required.

MODULES:

1. Video Capture Module

- **Purpose**: To continuously capture real-time video feed from the user's webcam.
- **Functionality**:Uses OpenCV `cv2.VideoCapture()` to access the system's default camera.Captured frames are flipped horizontally using `cv2.flip()` to provide a mirror view, making interactions more intuitive (like looking into a mirror).
- **Output**: Each frame is returned as a NumPy array representing the current view from the camera, flipped for user-friendly tracking.

2. Preprocessing Module

- **Purpose**: To prepare the video frame for color detection.
- **Functionality**:The captured BGR (Blue-Green-Red) image is converted to HSV (Hue-Saturation-Value) color space using cv2.cvtColor().HSV space is more effective for color-based segmentation because hue separates colors better than RGB.
- **Output**: An HSV version of the current frame, which is more robust for detecting specific colors like red.

3. Mask Generation Module

- **Purpose**: To isolate the red-colored object (the virtual pen) from the frame.
- **Functionality**:cv2.inRange() is used to create binary masks for two ranges of red (since red wraps around the hue circle).cv2.bitwise_or() combines these two masks into a single binary image.The resulting mask highlights only the areas of the image that match the color of the pen.
- **Output**: A binary mask where white pixels represent regions likely to be the red pe

4. Mask Refinement Module

- **Purpose**: To clean the binary mask by removing noise and enhancing the detected red region.
- **Functionality**: Applies morphological operations:
  - cv2.erode() removes small white noises.
  - cv2.dilate() enlarges the detected red region.
  - cv2.morphologyEx() with operations like "OPEN" or "CLOSE" helps fill holes and smooth boundaries.
- **Output**: A refined mask that better represents the actual location of the red pen.

5. Contour Detection Module

- **Purpose**: To locate the exact position of the red pen in the frame.
- **Functionality:** cv2.findContours() identifies all contour shapes in the mask. The largest or most relevant contour is considered as the red pen.cv2.moments() is used to compute the centroid (center of mass) of the detected contour. Optionally, cv2.minEnclosingCircle() can be used to draw a circle around the detected object.
- **Output**: Coordinates (x, y) of the red pen's tip in the current frame.

6. Tracking & Drawing Module

- **Purpose:** To record the path of the red pen and draw it onto a virtual canvas.
- **Functionality**: Uses a deque (double-ended queue) from the collections module to store a list of pen coordinates. Loops through the deque and uses cv2.line() to draw lines connecting the stored points. This simulates real-time drawing based on the movement of the pen.
- **Output**: Continuous drawing on the canvas as the red pen moves in front of the webcam.

7. Canvas Management Module

- **Purpose**: To maintain a persistent drawing canvas that holds the strokes of the virtual pen.
- **Functionality**: Initializes a blank canvas (paintWindow) using np.zeros_like() or similar. Updates this canvas with each new drawing point. Provides a mechanism to clear the canvas when the user presses 'c'.
- **Output**: An updated blackboard-style canvas that accumulates the user's drawings over time.

8. Display Module

- **Purpose:** To provide real-time visual feedback by showing the processed outputs to the user.
- **Functionality**: Uses cv2.imshow() to display three windows:
  - Live feed with drawing overlay
  - Binary mask view showing detected red regions
  - Drawing canvas to see the accumulated strokes
- **Output**: A visual interface where the user can see themselves, the tracking logic, and their virtual drawing.

9. User Interaction Module

- **Purpose**: To allow the user to interact with the system through simple keyboard controls.
- **Functionality**: cv2.waitKey(1) listens for keypress events. When 'q' is pressed, the application exits. When 'c' is pressed, the canvas and point history are cleared, giving the user a fresh slate.
- **Output**: Dynamic system behavior based on user inputs.

10. Exit & Cleanup Module

- **Purpose**: To gracefully terminate the application and release resources.
- **Functionality**: cap.release() stops the video capture and releases the camera.cv2.destroyAllWindows() closes all OpenCV GUI windows that were opened during runtime.
- **Output**: A clean shutdown of the application without any resource leaks or hanging windows.

LIBRARIES

1.NumPy

NumPy is a fundamental library in Python used for numerical operations. In this project, it is responsible for handling array-based data and creating the underlying structure of the drawing canvas. It helps define the size and shape of the blank image that serves as the virtual blackboard. Additionally, NumPy is used to generate the kernel matrix required for morphological operations, which are essential for cleaning up the binary mask and improving object detection. Overall, NumPy facilitates efficient matrix manipulations that are critical for image processing tasks.

2. OpenCV (cv2)

OpenCV is the core computer vision library used in this system. It manages all visual processing tasks, starting from capturing live video feed through the webcam to rendering the final drawings. The frames from the webcam are flipped to provide a mirror view, making interaction more intuitive. These frames are then converted from one color format to another—specifically from BGR to HSV—because HSV provides better performance in detecting specific colors under varying lighting conditions.OpenCV is also used to define the color range for detecting the red-colored stylus and to generate a binary mask that isolates red objects from the background. It includes built-in morphological operations like erosion, dilation, and opening, which help remove noise and refine the mask. After obtaining a clean mask, contour detection is performed to identify the boundaries of the stylus. The largest detected contour is assumed to be the stylus tip, and its center point is calculated for tracking.This center point is used to draw on both the real-time video feed and the virtual canvas. OpenCV provides tools to display multiple windows for live tracking, the mask, and the canvas. Additionally, it manages keyboard interactions, allowing users to clear the canvas or exit the program. Finally, it handles the release of hardware resources and closes all visual windows at the end of the session.

3. Collections (deque)

The collections module provides a specialized data structure called deque, which stands for double-ended queue. In the Air Canvas system, the deque is used to store and manage the history of the stylus's tracked positions. This allows the program to remember where the user has drawn over time and connect those points with lines to form smooth strokes. Each time the stylus is lost or lifted, a new deque is initialized to start tracking a new stroke. This method is memory-efficient and ensures that old points are discarded automatically once the limit is reached. The use of deque makes the drawing experience fluid and organized, supporting multiple disconnected lines on the canvas.

ALGORITHMS

The **Air Canvas** system uses a combination of computer vision algorithms and techniques. While the implementation is relatively straightforward, it still incorporates several important **image processing and object tracking algorithms**. Here's a breakdown of the **algorithms used in the code**:

1. Color Detection using HSV Threshold

Algorithm Type: Color segmentation
Purpose: To detect a specific color (in this case, red) in a video frame.
How it works:
The frame is converted from the BGR color space to HSV (Hue, Saturation, Value), which allows better handling of lighting variations. Two separate HSV ranges are defined to capture the red hue .

A binary mask is created where pixels matching these color ranges are highlighted (white), and all others are blacked out.

2. Morphological Operations

Algorithm Type: Image preprocessing
Purpose: To clean up the binary mask by removing noise and small artifacts.

Techniques used:

- **Erosion**: Removes small white noise.
- **Opening**: Erosion followed by dilation to smooth object edges.
- **Dilation**: Expands the white region to recover object size lost during erosion.

These operations refine the mask to ensure better contour detection.

3. Contour Detection

Algorithm Type: Shape detection
Purpose: To find the outlines (contours) of detected colored regions in the binary mask.

How it works:
Contours are extracted from the cleaned-up binary image. The algorithm finds continuous curves or boundaries that surround white blobs (the red stylus in this case). The largest contour is assumed to be the stylus tip.

4. Object Tracking via Centroid Calculation

Algorithm Type: Moment-based tracking
Purpose: To track the position of the stylus from frame to frame.

How it works:
Using image moments (statistical properties of the contour), the centroid (center) of the detected object is calculated. This gives the precise position of the stylus tip, which is used for drawing.

5. Point History Tracking with Deque

Algorithm Type: Data structure-based tracking
Purpose: To remember the path the stylus has taken over time.

How it works:
A deque (double-ended queue) stores the sequence of centroids detected over time. These points are connected using lines to simulate drawing on the screen. When the stylus disappears, a new deque is started to separate strokes.

6.Virtual Drawing

Algorithm Type: Rendering algorithm
Purpose: To visualize the motion of the stylus as continuous strokes.

How it works:
Consecutive points from the deque are joined using straight lines on both the real-time camera feed and a blank canvas. This gives the illusion of drawing in mid-air.

RESULT

The Air Canvas system delivers a compelling demonstration of real-time, contactless drawing by leveraging computer vision techniques to track a colored object—specifically, a red-tipped stylus—in a webcam feed. When the program is run, it launches three key display windows that showcase different aspects of the system's operation. The first and most interactive window, labeled **"Tracking,"** shows the live video stream captured from the webcam. This window overlays a circular marker on the red object detected in the frame, indicating that the system is successfully tracking it. As the user moves this object through the air, the system continuously calculates the centroid of the object and connects those points with lines. This results in the appearance of digital "strokes" being drawn in mid-air, mimicking traditional hand-drawing movements.

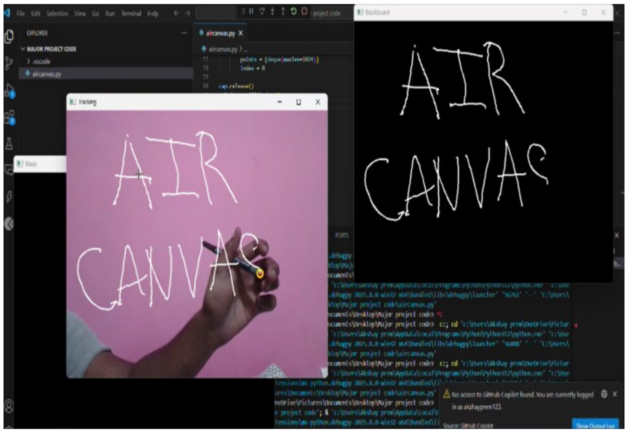
The second window, titled **"Blackboard,"** acts as a dedicated virtual drawing canvas. Unlike the tracking window, it does not show the webcam feed but only the strokes being drawn. This allows users to focus solely on their drawing output without any visual distractions. The strokes drawn on this blackboard persist until explicitly cleared, providing a clean and cumulative representation of the user's artwork or input. This canvas serves as the system's main output interface, and is particularly useful in applications like virtual whiteboarding, digital art, or interactive teaching environments.

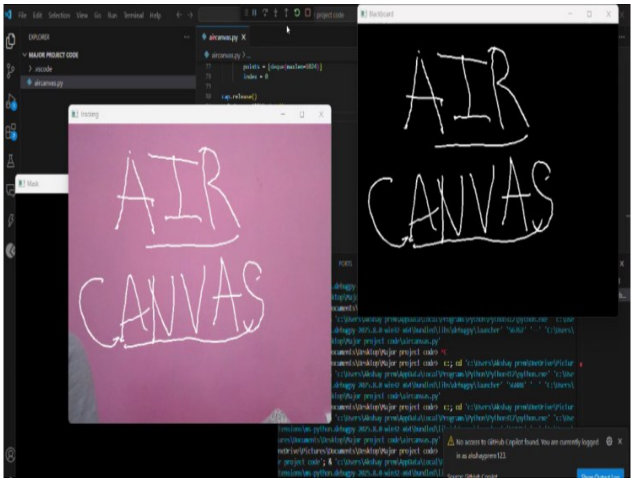
The third display, known as the **"Mask"** window, is a diagnostic tool that shows a binary (black and white) version of the processed frame. This mask highlights only the regions in the frame that match the predefined HSV (Hue, Saturation, Value) color thresholds for red. The white areas in the mask indicate the parts of the image where the red object is detected. This is critical for understanding the effectiveness of the color detection algorithm, especially under different lighting conditions or when the red object is partially obstructed.

To enhance usability, the system supports two simple keyboard interactions: pressing the **'c'** key clears the canvas by resetting all stored drawing points and refreshing the blackboard, allowing users to start a new sketch from scratch. Pressing the **'q'** key gracefully exits the program by closing all active windows and releasing the webcam resource. These interactive controls provide a user-friendly and efficient way to manage the virtual drawing session.

In terms of performance and user experience, the system delivers real-time tracking with minimal latency, allowing for smooth and continuous drawing even during relatively fast hand movements. The use of the HSV color model ensures robustness against changes in lighting, while morphological operations clean up the binary mask to improve accuracy in object detection. The integration of these image processing techniques enables the system to offer an intuitive, hygienic, and touch-free method of digital interaction.

Overall, the results reflect a successful implementation of an air-based drawing system that effectively combines object detection, gesture tracking, and real-time rendering. It provides a novel interaction model suitable for various domains such as education, art, accessibility tools, and human-computer interaction research. The current system also lays the groundwork for future improvements, such as multi-color or multi-object tracking, gesture recognition, pressure sensitivity, and even integration with augmented or virtual reality environments to enhance the overall interactive experience.





CONCLUSION:

The Air Canvas project presents a significant advancement in the domain of natural and contactless human-computer interaction by utilizing real-time computer vision techniques. It introduces an innovative way to draw digitally without any physical contact, simply by tracking the movement of a red-colored object, such as a pen cap, in front of a webcam. The core concept relies on converting each frame of the live video feed into the HSV (Hue, Saturation, Value) color space, which is more effective than the traditional RGB model for isolating specific colors under varying lighting conditions.

This color segmentation process, particularly tuned to detect red hues, enables the system to accurately identify and isolate the drawing object from the background.

To ensure the red object is clearly recognized, the system employs two separate HSV ranges to account for the unique wrap-around nature of red in the HSV spectrum. Once the target object is detected, a series of image processing steps—such as erosion, dilation, and morphological opening—are applied to remove noise and enhance the clarity of the detection. Contour detection is then used to identify the object's shape, and the system calculates the centroid of the largest contour, assuming it to be the tip of the user's stylus. This point is then tracked over time to map the movement of the hand.

To draw smooth and continuous lines, the program uses a data structure called a deque (double-ended queue), which efficiently stores the recent positions of the stylus. These stored points are connected using line-drawing functions, allowing the user to create clear and fluid digital sketches. The output is displayed on two separate windows: the "Tracking" window shows the live webcam feed along with the tracked path of the red object, while the "Blackboard" window serves as a clean virtual canvas where the drawing is persistently rendered. A third window, "Mask", displays the binary version of the frame where only the detected red object is highlighted, which is helpful for debugging or understanding how well the color segmentation is working.

User interaction with the system is simple and intuitive. By pressing the 'c' key, the canvas can be cleared instantly, enabling a fresh start. Pressing the 'q' key exits the application gracefully. These basic controls make the system highly accessible, even to non-technical users.

Despite its success, the system is not without limitations. It performs best in controlled environments with consistent lighting. Variability in light, background clutter, or other red objects in the frame can confuse the tracking algorithm, leading to inaccurate strokes.

However, these shortcomings also highlight opportunities for future development. Enhancements such as multi-color tracking, gesture-based command inputs, depth sensing, or integration with augmented reality (AR) and virtual reality (VR) platforms could greatly expand the functionality and application scope of the system.

In conclusion, the Air Canvas project showcases a practical, low-cost solution for creating a touchless, real-time drawing interface. It demonstrates how accessible technologies like webcams and open-source libraries (such as OpenCV) can be combined with efficient algorithms to build interactive tools that are both creative and hygienic.

Its potential applications are wide-ranging—from digital art and remote teaching to assistive technologies for people with physical disabilities. The project lays a robust foundation for future innovation in the field of gesture-based interaction and provides a glimpse into the possibilities of more immersive and intuitive computing environments.

REFERENCES:

[1] Kiran Ande, Anushka Korde, Deepak Dhale, Kartik Gujarkar, Hrutik Hiwase, Department of Computer Engineering, India. *"AIR-CANVAS: A Computer Vision Based Gesture Drawing System."* Published in **International Journal for Research in Applied Science & Engineering Technology (IJRASET)**, ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538, **Volume 13, Issue V, May 2025**.Scilit+2IJRASET+2IJRASET+2

[2] Mr. Pradeep , Mr. Ashwani Shivam , Mr. Tushar Sharma , Mr. Rajat Singh, Department of Information Technology, AISSMS Polytechnic, Pune, Maharashtra, India. *"Air-Canvas: Real-Time Gesture Recognition."* Published in **International Journal of Research Publication and Reviews (IJRPR)**, **Volume 6, Issue 4, April 2025**.IJRPR+1IJRPR+1

[3] Kiran Ande, Anushka Korde, Deepak Dhale, Kartik Gujarkar, Hrutik Hiwase, Department of Computer Engineering, India. *"AirCanvas using OpenCV and MediaPipe."* Published in **International Journal for Research in Applied Science & Engineering Technology (IJRASET)**, ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538,

[4] Mr.Prabakaran R, P.Sivaprakash, K Arun Kumar, Revathi A, Department of Computer Engineering, India. *"AIR CANVAS HAND TRACKING."* Published in **Journal of Emerging Technologies and Innovative Research (JETIR)**.Jetir

[5] Prof. Supriya Telsang, Rajkumar Dongre, Siddhi Rajeshirke, Kundan Rajput, Rajendra Rajpurohit, Manish Rakhewar, Tejas Rajguru, Department of Computer Engineering, India. *"Virtual Drawing Board (Drawing and Writing Using Hand Gestures)."* Published in **International Journal for Research in Applied Science & Engineering Technology (IJRASET)**, ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538.IJRASET

[6] Dr M. Srinivasa Sesha Sai , Singam Sai Prakash Reddy, Vuyyuru Chaitanya Praveen Reddy, Pedaprolu Sai Kumar, Nallaka Naga Siva Sai5 , Department of Computer Engineering, India. *"Expressive Digital Paint / Writing Using Mediapipe & OpenCV."* Published in **International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), March 2024**.IJIRSET

[7] Mr. Deven Gupta, Ms. Anamika Zagade, Ms. Mansi Gawand, Ms. Saloni Mhatre, Prof. J.W. Bakal, Department of Computer Engineering, [Institution Name], India. *"Gesture Controlled Virtual Artboard."* Published in **International Journal for Research in Applied Science & Engineering Technology (IJRASET)**, ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538.IJRASET+1IJRASET+1

[8] Soham Pardeshi, Madhuvanti Apar, Chaitanya Khot, Atharv Deshmukh, Department of Computer Engineering, India. *"Air Doodle: A Realtime Virtual Drawing Tool."* Published in **International Journal for Research in Applied Science & Engineering Technology (IJRASET)**, ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538.