PROJECT BASED LEARNING REPORT

On

AIR CANVAS

Submitted by:

Team Members:

1. LAKSHAY AGGARWAL[1/23/SET/BCS/562]

2. SOUMAY KOHLI [1/23/SET/BCS/565]

Under the Guidance of:

Faculty Name: Dr. GINNI SEHGAL

Department of Computer Science and Engineering

Manav Rachna International Institute of Research and Studies Academic Year: 2025–2026

ABSTRACT

AirCanvas is an AI-driven virtual painting system that allows users to draw in the air using simple hand gestures. Instead of relying on a mouse, stylus, or touchscreen, the system captures hand movements through a webcam and converts them into digital strokes in real time. Built using OpenCV and Mediapipe, the project demonstrates how gesture recognition and computer vision can be used to develop an intuitive, touch-free drawing interface.
The aim of this project is to explore the use of artificial intelligence in motion tracking, gesture interpretation, and real-time interaction, while providing a creative and accessible platform for digital expression.

INTRODUCTION

Artificial Intelligence has transformed how humans interact with machines, shifting from touch-based input to gesture-based control. The AirCanvas project is an example of this evolution, allowing the user to draw virtually by simply moving their hand in front of a camera.

The system uses Mediapipe's hand-tracking model to detect key landmarks on the hand and OpenCV to render the strokes on a digital canvas. This enables real-time drawing without any physical contact, making it useful for artistic applications, teaching, demonstrations, and assistive tools for users who cannot rely on traditional input devices.

This project helped us understand how AI algorithms interpret human gestures, how computer vision handles real-time processing, and how Python integrates multiple libraries to build functional interfaces.

OBJECTIVES

The major objectives of this project were:

- To design a touch-free drawing system using AI-based hand-gesture detection.
- To implement real-time hand tracking using Mediapipe.
- To create a virtual canvas using OpenCV for drawing strokes and shapes.
- To add controls for brush color, size, and erasing through gesture commands.
- To demonstrate the use of computer vision in user-interaction systems.

SYSTEM ANALYSIS

The traditional drawing tools require physical input devices, which limit natural interaction. AirCanvas aims to address these limitations by making drawing more intuitive and accessible.

Key Points of System Analysis

- The system relies on a webcam as the primary input device.
- Mediapipe provides 21 precise hand landmarks that help identify gestures such as pointing, selecting, or erasing.
- OpenCV handles frame capture, canvas generation, and rendering strokes smoothly.
- The system processes each frame in real time, so efficient gesture detection and minimal latency are essential.
- No external dataset is required because Mediapipe uses pre-trained models.

SYSTEM DESIGN

Hand Gesture Design

- **Index finger up:** Draw on the canvas.
- **Index + middle finger up:** Select colors, brush size, eraser, or save options.
- **Closed fist:** Acts as an eraser mode.
- **Open palm:** Pause/reset.

Software Architecture

1. **Input Layer** – Webcam captures real-time frames.
2. **Processing Layer** – Mediapipe detects hand landmarks and interprets gestures.
3. **Drawing Engine** – OpenCV updates the canvas based on detected gestures.
4. **UI Layer** – Provides swatches, options, and tool indicators.

Technology Stack

Component	**Description**
Programming Language	Python
Libraries	OpenCV, Mediapipe, NumPy
IDE	IDLE / VS Code
OS	Windows 11
Hardware	Laptop/PC with Webcam

IMPLEMENTATION

The project was implemented using Python along with OpenCV for image processing and Mediapipe for gesture recognition. The implementation included:

Major Modules

1. **Hand Tracking Module**
    - Detects 21 hand landmarks.
    - Identifies finger positions to determine gestures.
2. **Gesture Interpretation Module**
    - Converts finger combinations into commands (draw/select/erase).
3. **Drawing Canvas Module**
    - Creates a transparent canvas layer.
    - Draws strokes using fingertip coordinates.
4. **User Interface Module**
    - Displays color palette, brush size controls, eraser, clear, and save options.
    - Handles screen overlays for a clean user experience.
5. **Saving and Clearing Options**
    - Artwork can be saved as an image file.
    - Canvas can be cleared at any time.

Code Structure

Although the complete code is lengthy, it mainly consists of:

- Initializing webcam and UI
- Running Mediapipe's hand detector
- Tracking gestures
- Drawing strokes using OpenCV
- Implementing buttons for color, size, eraser, clear, and save

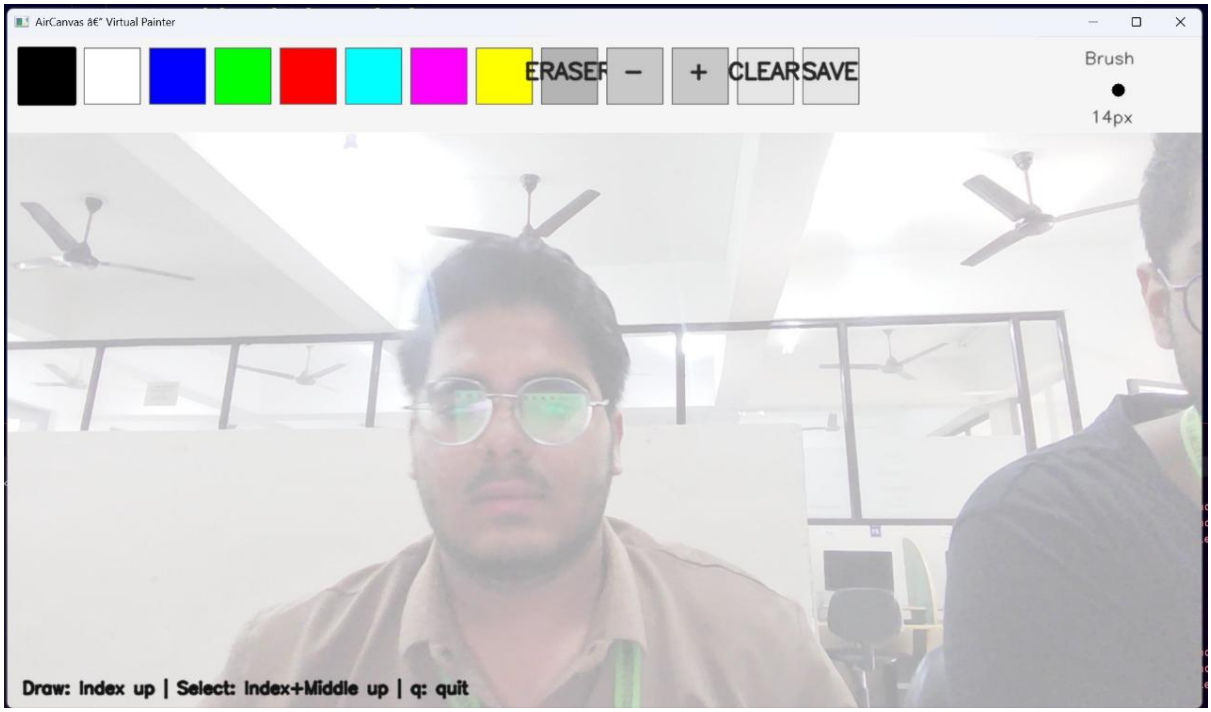
RESULTS AND DISCUSSION

AirCanvas performs consistently well under normal lighting and allows smooth drawing with hand gestures. The system successfully converts fingertip motion into digital strokes with minimal delay.

Observed Results

Feature	Observation
Hand Tracking	Stable up to ~2–2.5 meters distance
Drawing Smoothness	Very low latency (~0.1s)
Color Selection	Works accurately
Eraser Mode	Removes strokes cleanly
Save Option	Exports artwork as .png

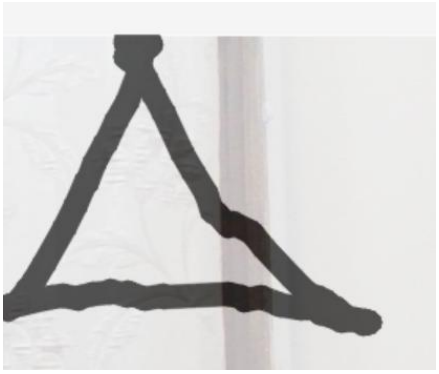
Sample Outputs



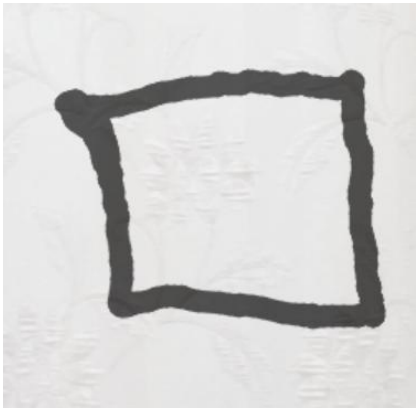
Users can draw shapes such as:

- Circles

- Triangles



- Squares



The shapes reflect the user's hand movement in real time.

CHALLENGES FACED

- Difficulty in accurate tracking under low lighting conditions.
- Distinguishing between similar gestures (like two-finger vs. three-finger).
- Reducing accidental selections when the hand moved across UI elements.
- Maintaining smooth drawing while handling frame-by-frame processing.

How We Resolved Them

- Adjusted tracking confidence thresholds.
- Added delay logic (debouncing) to reduce accidental selections.
- Optimized frame handling to improve drawing smoothness.
- Improved the UI layout to reduce gesture misinterpretation.
- Used different versions of mediapipe and python.

CONCLUSION AND FUTURE WORK

The AirCanvas project successfully demonstrates how AI, computer vision, and gesture recognition can work together to create an interactive, touch-free virtual drawing environment. It provides a fresh way to express creativity without traditional input devices and highlights practical applications of real-time hand tracking.

Future Enhancements

- Adding multi-hand drawing for collaborative sketches.
- Integrating voice commands for tool switching.
- Using AR glasses to draw in augmented environments.
- Improving gesture recognition with custom models.

Overall, this project helped us practically apply AI, Python programming, and computer vision concepts while building something interactive and enjoyable.

REFERENCES

1. Mediapipe Documentation — Google Developers
2. OpenCV Official Documentation
3. NumPy Library Documentation
4. Various online tutorials and research articles on gesture recognition