Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi, Accredited by NAAC, with 'A' Grade)

Near Jnana Bharathi Campus, Bengaluru – 560056

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

A Mini-Project Report on

"AIR CANVAS USING MACHINE LEARNING"

Submitted in partial fulfilment of the requirement for the award of the Degree of

Bachelor of Engineering
in
Computer Science & Engineering

Submitted by

Basit Zahoor	**1DA20CS028**
Ananya B Narayan	**1DA20CS015**

For the academic year 2023-24

Under the Guidance of

Dr. Leena Giri G.

Associate Professor, Dept. of CSE,
Dr.AIT, Bengaluru – 56.

Visvesvaraya Technological University

Jnana Sangama, Belagavi, Karnataka 590018.

Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi, Accredited by NAAC, with 'A' Grade)

Near Jnana Bharathi Campus, Bengaluru – 560056

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Certificate

Certified that the Mini-Project work titled **"AIR CANVAS USING MACHINE LEARNING"** is a bonafide work carried out by **Basit Zahoor** and **Ananya B Narayan** bearing the USNs **1DA20CS028** and **1DA20CS015**, students of **Dr. Ambedkar Institute of Technology, Bengaluru,** in partial fulfilment for the award of **Degree in Bachelor of Engineering in Computer Science & Engineering** during the academic year 2023-24. It is certified that all corrections/suggestions indicated during Internal Assessment have been incorporated in the Mini-Project report deposited in the department. The Mini-Project report has been approved as it satisfies the academic requirements in respect of Mini-Project work prescribed for the said Degree.

Signature of the Guide
Dr. Leena Giri G.,
Associate Professor,
Department of CSE,
Dr.AIT, Bengaluru -56.

Signature of the HOD
Dr.Siddaraju,
Professor and HEAD,
Department of CSE,
Dr.AIT, Bengaluru – 56.

Signature of the Principal
Dr. Meenakshi M,
Principal,
Dr.AIT, Bengaluru – 56.

Viva-Voce Examination

Name of the Examiners Signature with Date

1.
2.

ACKNOWLEDGEMENTS

The sense of jubilation that accompanies the successful completion of this Mini-Project would be incomplete without mentioning and thanking all the people who played a vital role in the completion of this project by providing endless encouragement and support.

We would like to thank **Dr. Meenakshi M,** Principal, Dr.AIT, who has always been a great source of inspiration while carrying out this Mini-Project.

We am extremely grateful to **Dr.Siddaraju**, Professor and HEAD, Department of CSE, Dr.AIT for providing us constant encouragement and permitting us to utilize the required laboratory facilities and a congenial working environment for the successful completion of this Mini-Project.

We are highly indebted to my guide **Dr. Leena Giri G., Associate Professor**, Department of CSE, Dr.AIT for constant guidance and support, as well as for providing necessary information regarding the Mini-Project.

We would also like to thank all the teaching and non-teaching staff members of Department of Computer Science & Engineering Department for their support during the course of this Mini-Project implementation.

Lastly, we would like to thank our parents and friends whose constant encouragement and support was crucial in execution and completion of this Mini-Project.

Name of students
Basit Zahoor
Ananya B Narayan

ABSTRACT

The "AIR-CANVAS WITH ML" project intends to construct a system that makes use of machine learning (ML) algorithms to produce a virtual canvas in mid-air, enabling users to engage with digital information without the use of physical touch or conventional input devices.The system tracks the user's hand movements and gestures in real-time using computer vision techniques, allowing users to draw, write, or control virtual objects in the air. The recorded hand movements are analysed and translated by machine learning (ML) algorithms into digital gestures or commands that are displayed on a screen or projected in augmented reality (AR).To increase the system's responsiveness and accuracy, the project integrates machine learning algorithms with cutting-edge computer vision techniques.The system improves at recognising varied gestures, comprehending various sketching styles, and offering a smooth user experience by training the ML models on large datasets of hand movements. AIR-CANVAS WITH ML has a wide range of possible applications, from creative expression and design to interactive presentations and immersive gaming experiences. It provides consumers with a fresh and simple way to interact with digital content, encouraging innovation and interaction in virtual spaces.The project focuses on user experience design and optimisation to make sure the system is user-friendly, ergonomic, and responsive in addition to exploring the technical aspects of machine learning and computer vision. The AIR-CANVAS WITH ML project seeks to push the boundaries of human-computer interaction and open up fresh avenues for creative expression and interactive digital media.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

Machine Learning is a branch of Artificial Intelligence that aims at solving real-life engineering problems. This technique requires no programming, whereas it depends on only data learning where the machine learns from pre-existing data and predicts the result accordingly. Machine Learning methods have benefit of using decision trees, heuristic learning, knowledge acquisition, and mathematical models.

Air Canvas is an innovative concept that combines machine learning (ML) techniques with computer vision to create a virtual board. It offers users a natural and intuitive way to interact with digital content by using hand movements and gestures. The underlying ML algorithms and computer vision technology enable the system to track and interpret these movements, allowing users to draw, manipulate, and interact with virtual objects without the need for physical drawing tools. The primary objective of Air Canvas is to provide a seamless and immersive user experience, bridging the gap between the physical and digital worlds.

1.1 Existing System

Existing systems for digital art creation and drawing predominantly rely on physical touch interfaces, like tablets or styluses, which can constrain artists' mobility and precision. These limitations become apparent when attempting to create art in diverse settings or on the go. Moreover, conventional touch-based interfaces may not offer the same level of innovative interaction that drawing in the air can provide. The freedom to express creativity through hand gestures and motion tracking technology opens up new possibilities for artists, designers, and creators, offering a more immersive and intuitive canvas that transcends the confines of physical devices. This shift towards air canvas systems promises to redefine digital art creation, making it more versatile, interactive, and adaptable to various creative contexts.

1.2 Proposed System

The aim is to develop an AI-powered air canvas system that enables users to create digital artwork through hand gestures in the air. The system should utilize computer vision techniques to track the user's hand movements and convert them into drawings on a virtual canvas. It should provide real-time feedback, allowing users to select colors, sizes, and erase mistakes using intuitive gestures. The AI model should be capable of recognizing different shapes and patterns, enhancing the user's drawing experience.

1.3 Objective of the Project

The objective of the air canvas using machine learning is to create a dynamic and interactive platform that enables users to express their creativity by drawing and painting in the air using hand gestures. By harnessing the power of machine learning and computer vision, the system aims to accurately capture and interpret these gestures, translating them into digital artwork on a virtual canvas. This innovative application seeks to provide an intuitive and engaging user experience, promote artistic expression, and showcase the potential of machine learning in interactive and creative domains. Additionally, the system could serve as an educational tool, helping users of all ages explore the fusion of technology and art while fostering a deeper understanding of gesture recognition and machine learning concepts.

1.4 Design Methodology

Designing an air canvas using machine learning involves :

1. Define Objectives and Requirements : Clearly outline the goals of your air canvas system, such as real-time gesture recognition and dynamic digital art creation. Specify the hardware and software requirements, including the choice of cameras, sensors, and machine learning frameworks.

2. Data Collection : Gathers a diverse dataset of hand or stylus movements to train your machine learning model. Capture a wide range of gestures, strokes, and styles to ensure the model's robustness.

3. Preprocessing : Preprocess the collected data, including image calibration, noise reduction, and frame alignment. Extract relevant features from the data, such as hand position, speed, and direction.

4. Machine Learning Model Selection : Choose appropriate machine learning algorithms, such as Convolutional Neural Networks (CNNs) for image recognition and recurrent neural networks (RNNs) for sequential data processing. Design and train models to recognize gestures and interpret them as digital brush strokes.

5. Real-time Tracking : Implement computer vision techniques to track the artist's hand or stylus in real-time using cameras or sensors. Integrate the trained machine learning model to interpret gestures and strokes as input.

6. Digital Art Generation : Develop algorithms and techniques for translating recognized gestures into digital art on a virtual canvas. Consider factors like brush size, color, and stroke dynamics to create expressive and responsive artwork.

Chapter 2

LITERATURE SURVEY

1. “Finger Detection” by Izane : Focuses on computer vision techniques for identifying and tracking fingers in real-time. It delves into the use of image processing algorithms to detect and locate fingers within video frames or image data. The study explores applications in various fields, including human-computer interaction and gesture recognition. Izane's work contributes to the development of more accurate and responsive finger-tracking systems, enabling advancements in interactive technologies.
2. “Finger Detection and Tracking” by Amar Prakash Pandey : Tracking the movement of a finger is an important feature of many computer vision applications. In this application, A histogram based approach is used to separate out the hand from the background frame. Thresholding and Filtering techniques are used for background cancellation to obtain optimum results.
3. "Create Air Canvas using Python-OpenCV" by InfoAryan : OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it is integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image patterns and their various features we use vector space and perform mathematical operations on these features.
4. “Drawing in PyGame” by pygame v2.5.0 documentation : How to draw several simple shapes to a surface is instructed. Most of the functions take a width argument to represent the size of stroke (thickness) around the edge of the shape. If a width of 0 is passed the shape will be filled (solid). All the drawing functions respect the clip area for the surface and will be constrained to that area. All the drawing functions accept a color argument that can be one of the following formats: A pygame.Colorpygame object for color representations object is an (RGB) triplet (tuple/list) and (RGBA) quadruplet (tuple/list).
5. “Guide to Set Up Pi Camera & Some OpenCV Image Processing Basics” by Xitang Zhao: Air Canvas is a hands-free digital drawing canvas that utilizes a Raspberry Pi, a PiCamera, and OpenCV to recognize and map hand gestures onto a PiTFT screen.The user’s “brush” can be modified in size and color by using built-in buttons. The direction of the brush is controlled completely using open source OpenCV software and modified to map the pointer finger onto the screen using Pygame for a calibration screen to record the color of the user’s hand.

Chapter 3

REQUIREMENTS SPECIFICATION

System requirements are the prerequisites that are often used as a guideline as opposed to an absolute rule. Requirements can be classified as functional requirements, nonfunctional requirements, software requirements and hardware requirements.

3.1 Functional Requirements

1. Gesture Recognition and Mapping: The system must accurately recognize a variety of hand gestures, such as strokes, taps, and swipes, and map them to corresponding canvas actions like drawing lines, changing colors, and erasing.

2. Real-time Canvas Interaction: Users should experience seamless real-time interaction between their hand gestures and the virtual canvas, ensuring that their movements translate fluidly into visual elements on the screen.

3. Dynamic Tool Selection and Parameter Control: The system should allow users to dynamically switch between different drawing tools (brushes, erasers, etc.) and adjust parameters (brush size, opacity, color) using intuitive gestures, enhancing their control over the creative process.

4. Artistic Assistance and Generation: The AI should offer creative assistance by generating artistic elements based on user gestures, allowing for features like auto-completion of shapes, style transfer, and the creation of intricate patterns.

5. User-friendly Interface and Feedback: The user interface should be intuitive, providing clear visual feedback on selected tools, colors, and canvas actions, ensuring that users understand the system's response to their gestures and actions.

3.2 Non-Functional Requirements

1. Performance: The system should provide real-time and responsive drawing capabilities, ensuring smooth and low-latency interaction between the user's gestures and the virtual canvas.

2. Portability: The application should be accessible and function consistently across a variety of devices and platforms, including desktop computers, tablets, and smartphones.

3. Reliability: The system must maintain stable and consistent performance over extended periods, without crashes or data loss, to ensure a reliable drawing experience.

3.3 Hardware Requirements

1. Computer System: A computer system with sufficient processing power and memory to handle the ML algorithms and image processing tasks.
2. Camera: A high-resolution camera capable of capturing clear and detailed images of the user's hand gestures.
3. Display Device: A screen or projector to display the digital canvas and the artwork created by the user.
4. Tracking Device : If precise hand tracking is required, a tracking device like a depth sensor or a motion capture system can be utilized.

3.4 Software Requirements

1. Operating System: The project should be compatible with the chosen operating system (e.g., Windows, macOS, Linux).
2. Machine Learning Framework: Select an appropriate ML framework such as TensorFlow, PyTorch, or Keras to train and deploy ML models.
3. Computer Vision Libraries: Utilize computer vision libraries like OpenCV or Dlib to perform image processing tasks, hand detection, and tracking.
4. Programming Language: Choose a programming language such as Python or C++ to develop the project.
5. Development Environment: Select a suitable integrated development environment (IDE) for coding and debugging, such as PyCharm or Visual Studio Code.
6. Data Storage: Determine the method and format for storing and managing the training data and any pre-trained models used in the project.
7. Version Control: Employ a version control system like Git to manage the source code and collaborate with team members efficiently.
8. User Interface: Design an intuitive and user-friendly interface that allows users to interact with the Air Canvas system easily.
9. Dependency Management: Use a package manager like pip or conda to handle the installation and management of project dependencies.
10. Documentation: Create comprehensive documentation to explain the project's functionality, installation instructions, and usage guidelines.

Chapter 4

SYSTEM DESIGN

The system design for the air canvas using machine learning encompasses a multi-tier architecture. At the core is a real-time gesture recognition module employing deep learning algorithms, accurately capturing and classifying hand movements. This module communicates with a canvas interaction layer, which translates recognized gestures into corresponding actions on a virtual canvas through computer vision techniques. An AI-driven assistance component enhances creativity by generating artistic elements based on user gestures. The user interface layer provides an intuitive interface for tool selection, color picking, and real-time visual feedback. The system leverages efficient resource management to ensure seamless performance, with the potential for deployment across various platforms, fostering an engaging and dynamic user experience in the realm of interactive digital art.

4.1 System Architecture

The system architecture of the air canvas employs a modular design. At its core is a gesture recognition module, utilizing deep learning models to interpret hand movements. This module interfaces with a canvas interaction layer, translating gestures into actions on a virtual canvas via computer vision techniques. An AI assistance component augments creativity by generating art elements. The user interface provides tool selection, color control, and immediate visual feedback. The architecture ensures real-time responsiveness, scalability, and adaptability for deployment across diverse platforms, delivering an immersive and interactive artistic experience.

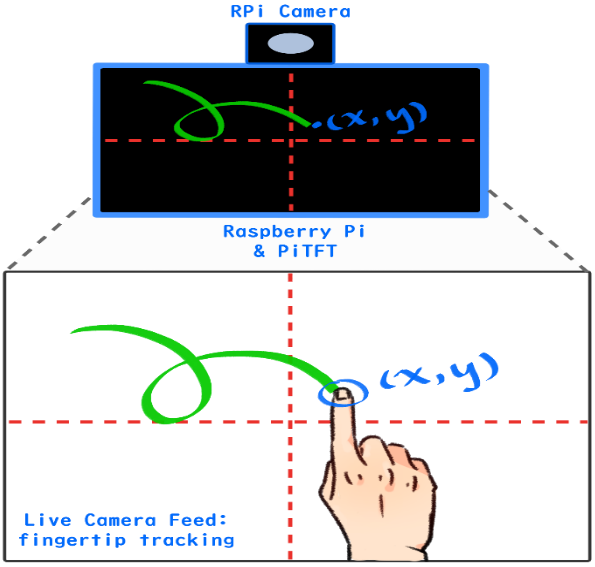

Figure 4.1.1 System Architecture

4.2 Use Case Diagram

A use case diagram typically consists of actors, use cases, and the relationships between them. Here's how it might look for the air canvas system:

Actors: User

Use Cases:

1. Draw on Canvas

2. Select Drawing Tool

3. Choose Color

4. Adjust Brush Size

5. Erase Mistakes

6. AI-generated Art Assistance

7. Save Artwork

8. Share Artwork

9. Change Canvas Background

Relationships:

- User initiates Draw on Canvas, Select Drawing Tool, Choose Color and Erase Mistakes use cases.
- User interacts with "AI-generated Art Assistance" for creative suggestions.
- User can "Save Artwork" and "Share Artwork" with others.
- User has the option to "Change Canvas Background."

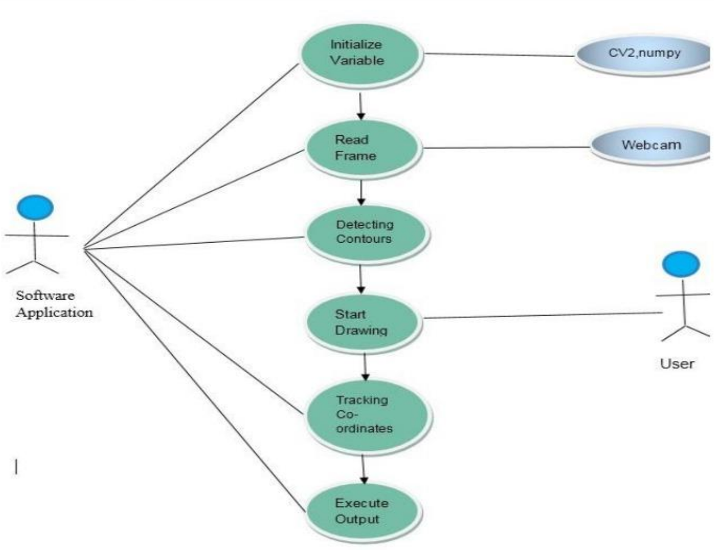

Figure 4.2.1 Use case diagram

Chapter 5

IMPLEMENTATION

In the implementation of the "Air Canvas" machine learning project, I utilized the power of the Mediapipe library to create a captivating and interactive drawing experience. By employing the hand tracking capabilities of Mediapipe, I enabled users to effortlessly paint and sketch in the air using their hand gestures. Mediapipe's hand tracking models provided real-time tracking of hand movements and fingertip positions. As users moved their hands in front of a webcam, the system translates their gestures into dynamic strokes on a digital canvas, allowing them to express their creativity freely.

1. Hand Tracking and Landmark Detection:

Mediapipe: The project heavily relies on the Mediapipe library, which provides pre-trained models for hand tracking and landmark detection. These models are capable of detecting hand positions and landmarks (such as fingertips) in real-time from webcam input.

2. Computer Vision:

OpenCV (cv2): The OpenCV library is used for image and video processing. It's utilized to capture webcam frames, perform image manipulations, draw shapes and text on the canvas, and display the final result.

3. Gesture Recognition and Interaction:

Gesture Detection: The code detects gestures based on the position of the user's hand and fingertips captured by the Mediapipe hand tracking model. Here when we show single finger its allows to “select tool”. If two fingers are shown then system allows to draw.

Hand Movement Interpretation: The code interprets the movement of the user's hand to select drawing tools and perform corresponding actions like drawing lines, rectangles, circles, and erasing.

4. Interaction Mapping:

Mapping Gestures to Actions: The code maps specific hand gestures to drawing tools and actions (line, rectangle, draw, circle, erase).

5. User Interface and Visualization:

Canvas Visualization: The script uses OpenCV to display a canvas on which the user can draw using their hand gestures.

Chapter 6

RESULTS

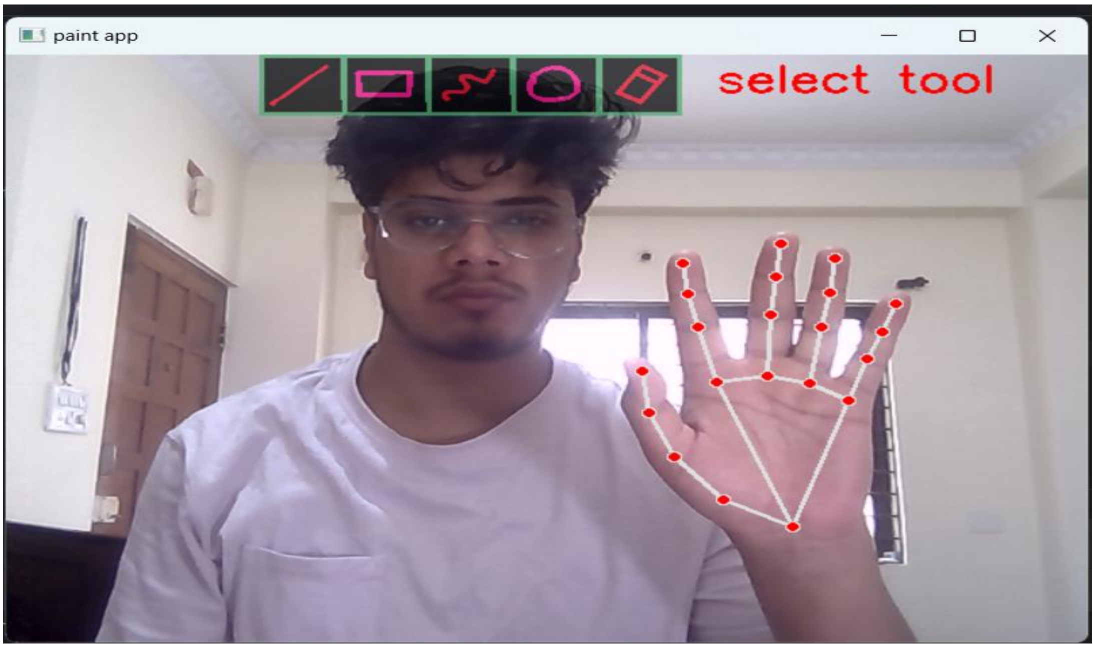

Figure 6.1 Hand Detection

The system uses computer vision and machine learning to detect and track the user's hand movements in real-time.

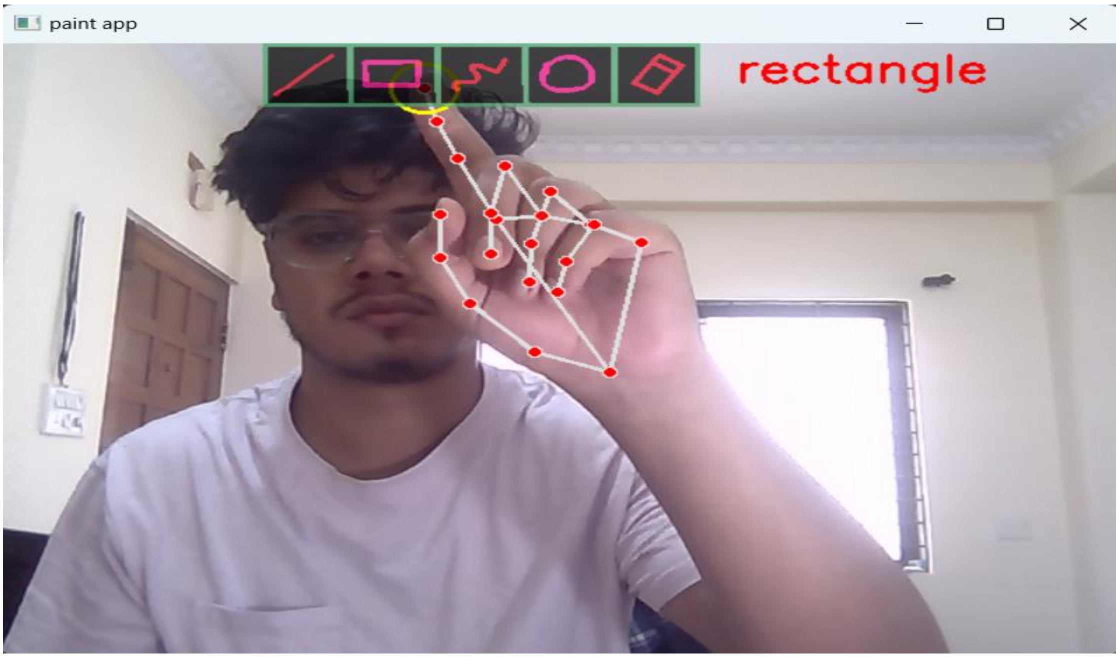

Figure 6.2 Select Tool

Allows users to outline and select specific drawn elements or objects on the canvas, enabling them to move, resize, or erase them easily, enhancing precision in editing and design.

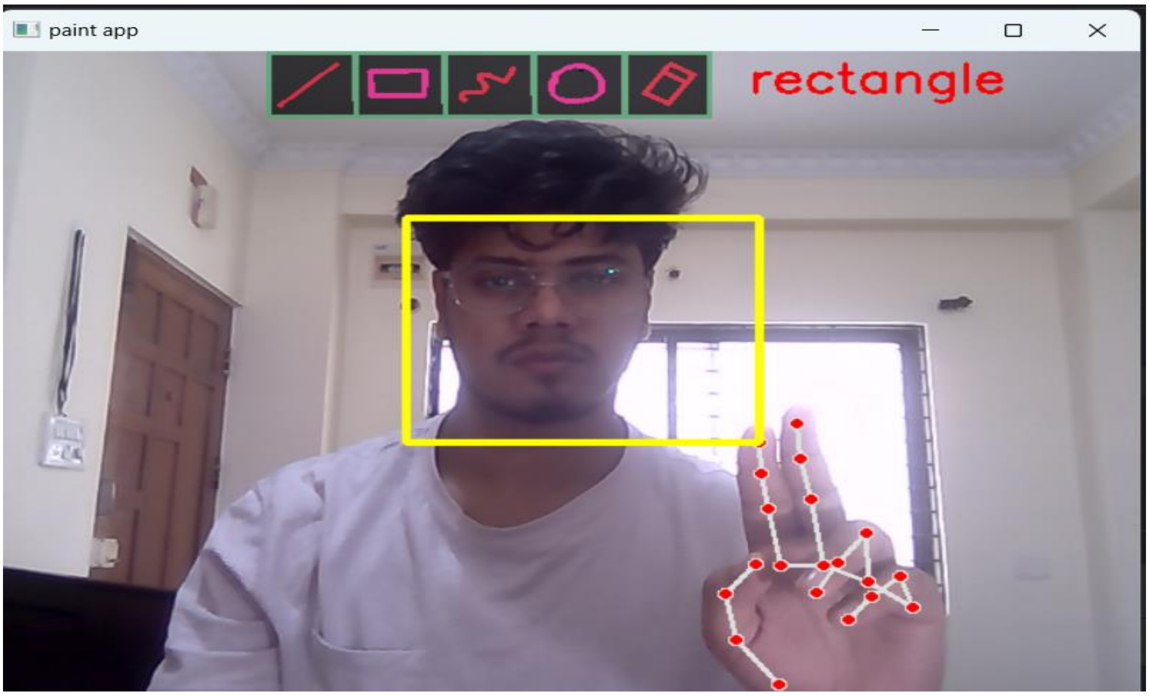

Figure 6.3 Draw

It interprets hand gestures to control drawing actions, enabling users to draw, erase, or manipulate the canvas without physical contact.

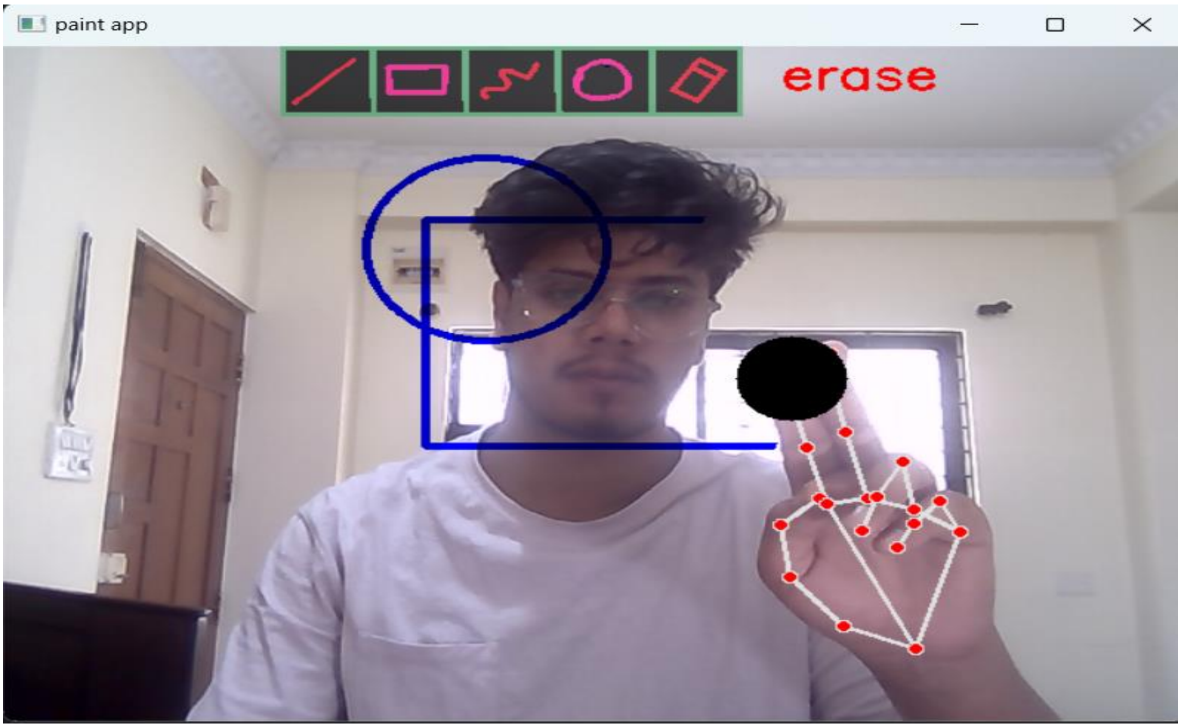

Figure 6.4 Erase

Enables users to remove or delete specific drawn lines or shapes by interacting with the canvas, providing a precise and convenient way to edit their drawings or designs.

APPLICATIONS

The Air Canvas project has the potential for application in various areas where interactive and creative digital experiences are desired. Some specific areas where the Air Canvas project can be applied include:

1. Education: Air Canvas can be utilized in classrooms and online learning environments as an interactive and collaborative tool. It allows teachers and students to engage in real-time drawing, diagramming, and annotation activities, fostering active learning and visual representation of concepts.

2. Business and Presentations: Air Canvas can enhance business presentations, brainstorming sessions, and team meetings. It provides a dynamic platform for visualizing ideas, creating diagrams, and collaborating on projects, enabling teams to work together effectively, whether they are in the same physical location or dispersed remotely.

3. Design and Architecture: Air Canvas can be a valuable tool for designers, architects, and artists. It allows them to sketch, annotate, and iterate on design concepts in a virtual environment. The shape recognition feature can assist in creating accurate diagrams and blueprints, enhancing the overall design process.

4. Remote Collaboration: Air Canvas facilitates remote collaboration by enabling multiple users to interact with a shared virtual whiteboard. This feature is particularly beneficial for teams working in different locations, allowing them to collaborate, share ideas, and make annotations in real time, improving productivity and communication.

5. Training and Simulation: Air Canvas can be used in training scenarios, such as medical simulations or engineering training, where learners can interact with virtual whiteboards to practice procedures, analyze data, and draw diagrams. It offers a dynamic and immersive learning experience.

6. Creative Arts and Entertainment: Air Canvas can be a platform for digital artists and performers to create interactive art installations, virtual performances, or live demonstrations. It allows them to express their creativity in real time, leveraging hand gestures and shape recognition for unique artistic experiences.

CONCLUSION AND FUTURE ENHANCEMENT

Conclusions:

In conclusion, The code effectively detects hand movements and translates them into various drawing tools by simply moving their hands in the air. The project showcases the power of computer vision and real-time gesture recognition in enabling intuitive and engaging user experiences.

Future Enhancements:

1. **Color Selection:** Implement the ability to select colors using specific hand gestures, providing users with a wider range of creative options.
2. **Drawing Modes:** By incorporating multiple drawing modes, such as freehand drawing, straight-line drawing, and curved-line drawing, each activated by distinct gestures.
3. **Undo and Redo:** Introduce undo and redo functionalities, allowing users to easily correct mistakes and experiment with different drawing ideas.
4. **Saving and Sharing:** Enable users to save their created artwork as image files and share them on social media platforms.
5. **Multi-User Support:** Extend the project to support multiple users simultaneously, enabling collaborative drawing experiences.
6. **Gesture Learning:** Implement machine learning techniques to improve gesture recognition accuracy over time, enhancing the responsiveness and accuracy of the application.
7. **Brush Customization:** Allow users to adjust brush size and opacity by utilizing hand gestures, providing greater control over their artistic creations.
8. **Backgrounds and Layers:** Integrate the ability to add backgrounds and work with multiple layers for more complex and detailed artwork.
9. **User Interface Improvements:** Develop a more intuitive and user-friendly graphical user interface (GUI) to help users navigate through tools and options seamlessly.
10. **Optimization:** Optimize the code for better performance, potentially leveraging hardware acceleration techniques or GPU processing to achieve smoother rendering and faster gesture recognition.