Air Canvas: Hand Tracking Using OpenCV and MediaPipe

Tamalampudi Hemaa Chandhan
Dept of CSE (Data Science)
Jain (Deemed-To-Be University)
Ramanagara District, Karnataka
20btrcd039@jainuniversity.ac.in

Kavin Kumar R
Dept of CSE (Data Science)
Jain (Deemed-To-Be University)
Ramanagara District, Karnataka
20btrcd028@jainuniversity.ac.in

Nalin Raj
Dept of CSE (Data Science)
Jain (Deemed-To-Be University)
Ramanagara District, Karnataka
20btrcd034@jainuniversity.ac.in

Neelam Nanda Kishore Reddy
Dept of CSE (Data Science)
Jain (Deemed-To-Be University)
Ramanagara District, Karnataka
20btrcd035@jainuniversity.ac.in

Dr.Mohammed Zabeeulla A N
Dept of CSE (Data Science)
Jain (Deemed-To-Be University)
Ramanagara District, Karnataka
z.mohammed@jainuniversity.ac.in

***Abstract*— Establishing a real-time system that uses a camera to recognise human actions and transform them into digital strokes on a canvas is the intended result of this research. This device tracks hand gestures using motion analysis techniques, and it recognises activities using deep learning algorithms. The biggest challenge is accurately detecting human activity and creating digital strokes in real-time while accounting for variations in appearance, motion, and camera angle. The Air Canvas technology enables artists to produce paintings or drawings using their natural hand movements, making it a more expressive and intuitive way to create digital art. Potential uses for this technology include digital art and computer-human interaction. Python is the programming language used to create this system. To track finger locations, a camera and Mediapipe are utilised. On the canvas or in the available space, forms are drawn using algorithms incorporated into the computer vision system. The system tracks the hand that serves both as an eraser and a pen to draw or make different forms. The technology being used in this study enables us to make drawings by merely using our hands. To overcome or minimise these limitations, we created this project, which makes use of innovative technology and straightforward techniques.**

Keywords—canvas, motion analysis, Mediapipe, Computer Vision, human action, digital art

I. Introduction

A. Overview

The field of human action recognition in an air canvas environment has gained significant attention in computer vision and human-computer interaction. This report focuses on the development of a system called Air Canvas, which utilizes computer vision techniques and machine learning algorithms to recognize and interpret finger gestures in real time on a virtual canvas.

B. Problem Definition

The problem addressed in this research is the recognition of human actions in real-time using a camera and the translation of these actions into digital strokes on a canvas. The main challenge is to develop a system that can accurately recognize various human actions, while considering variations in appearance, motion, and camera viewpoint. The system should be capable of generating digital brush strokes that closely resemble the natural hand movements of individuals. Overall, the problem is to bridge the gap between human actions, captured through a camera, and their translation into digital brush strokes, in real-time, with a focus on accuracy, adaptability to variations, and providing artists with a more intuitive way of creating digital art.

C. Objectives

- To create a camera-based system that can instantly detect and recognize hand gestures and movements.
- To create a digital canvas that users may draw on with their hands and fingers.
- Utilize motion analysis to track a person's hand movement, and translate it into digital strokes on a canvas.

II. Literature Survey And Related Work

Authors in [1] uses CNN techniques to first detect the fingertip by an RGB camera eventually the KCF tracker algorithm is used to convert the detected hand region into HSV color space. The key disadvantage is that the researchers compromised on the recognition of fingerprints by using basic algorithms and an elementary framework which affects the accuracy of the system in real-time making this project look downsized.

Authors [2] use a dataset made manually by them in several backgrounds for better prediction of a fingertip using the Single Shot Detector(SSD). Also, it used the RCNN model to train its system for faster and Regional Proposed Network(RPN) to detect the fingertip in the picture captured by OPENCV. An important backdrop is that even though the system is fast at processing and good at averaged accuracy; the presence of any red color in the background leads to false and error predictions while using.

Authors in [3] proposed the system using Mediapipe which helps to reduce the process of image processing to detect the position of fingers, which in detail first detects hand landmarks and obtain positions according to it and even usage

of Python Tkinter to open PDF files and annotate/edit by using Hand landmarks for detecting hand gives out a precise analysis of palm recognition compared to all the other existing methods/systems. The minimal improvement needed to answer is that in this proposed system, switching and performing the various availabilities always requires physical effort to change the modes like from Writing to Painting, depicting shapes instead of scribblings, Editing pdf, and Saving the work on canvas into an image.

The method presented in [4] used the Kinect sensor's depth and colour data to determine the hand's form. The process of gesture detection is still relatively difficult even with the Kinect sensor. There is not much resolution on this Kinect sensor. When monitoring a large object, like the human body, it works effectively. However, it's challenging to track something that's as small as a finger.

The technique explained by the authors in [5] involves placing an LED on the user's finger and tracking it with the web camera. The character which is stored in the database will be compared with the one that was drawn. The alphabet is given back which satisfies and appears identical. It requires installing a red LED light source focused on the finger. It is also necessary that the LED light is the only thing that is red in the web camera's field of view.

The researchers in [6] have developed a novel HCI (Human–computer interaction) approach that integrates a camera, an accelerometer, two Arduino microcontrollers, and an ultrasonic distance sensor. The primary idea behind this interface is to gather motion data using ultrasonic distance sensors. In order to record the motions, the distance between the hand and the distance sensor is recorded.

In [7], the researchers created a virtual paint application that follows the user's hand movements while they write on the screen using ball-tracking technology. As a perimeter, they've utilized a glove with a ping-pong ball strapped to it.

These related articles provide insightful knowledge in the fields of gesture recognition, motion analysis, and human action recognition. They aid with the comprehension of various methods, formulas, and applications in the context of identifying and deciphering human behaviors in real-time. By combining hand gesture detection, motion analysis, and digital art production, the proposed study on Air Canvas builds on these earlier efforts and advances the state-of-the-art in the related field.

III. Existing System

Color tracking is a technique used in computer vision to track the movement of an object based on its color in a given video feed. The technique involves detecting the pixels of a particular color within a frame and using this information to track the movement of the object in subsequent frames. This technique is commonly used for tracking hands in gesture recognition systems, where the user is required to wear an additional material, such as a bead or a finger glove, of a specific color to enable tracking. By using the color of the material as a reference, the system can distinguish between the object of interest (the user's hand) and the rest of the scene's background.

IV. Proposed System

In our proposed model, we utilize the Hand landmark model to track the entire hand and achieve accurate detection of hand coordinates within the detected hand regions. This allows us to precisely capture the movement and position of the user's hand. The model incorporates a gesture-based canvas, where different gestures correspond to different operations. For instance, if a single index finger is detected, the system enters the drawing/painting mode. This means that the user can freely move their finger on the air canvas, and the system will capture their movements and translate them into digital brush strokes on the canvas. This enables the user to create digital art by simply using their finger as a brush. In addition, if two fingers, specifically the index and ring fingers, are detected, the system switches to the selection mode. In this mode, the user can perform selection operations, such as choosing a specific element or area on the canvas. The system recognizes the gesture and gives the user the ability to interact with the digital content in a more accurate and stable manner.

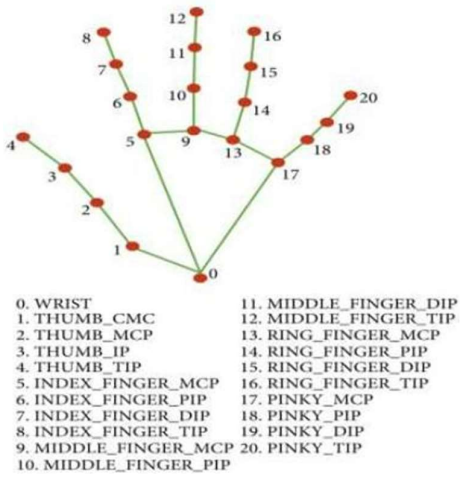

Fig 1: Hand landmarks

Our suggested approach offers a more natural and expressive manner for users to engage with the air canvas system by adding various motions and modes. It makes use of hand tracking and gesture recognition technology to let users manipulate the canvas or create digital art using their natural hand motions.

V. Methodology

A. Fingertip Detection Model:

Simple tools like an airbrush with distinctive colors are all that is needed to write in the air. The method makes use of a fingertip. We believe that people should be able to write in the air without needing to carry a piece of equipment with them. Deep Learning techniques were used to determine the fingertip in each frame, yielding a set of coordinates. We have employed pre-trained MediaPipe models to recognize hands and particular locations on them.

B. Get coordinates of fingertips on the screen:

The application constantly records the position of those landmarks on the screen and returns them. Based on the returned values we give the user the ability to change functionality. We use Tkinter to build a UI where the users can change modes by altering the coordinates of the landmarks on the screen.

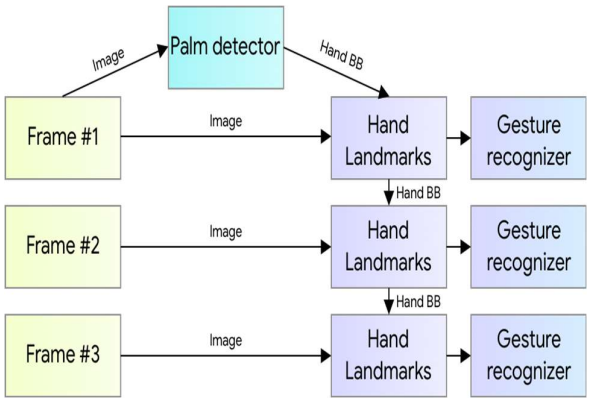

Fig 2: Program flow

C. Understand gestures:

The application is built in such a way that different gestures shown in front of the camera are detected and the corresponding actions are performed. Such gestures which are introduced in the application are,

1-finger up – Drawing mode

2-fingers up – Selection mode

All fingers up – Clear all

VI. ALGORITHM

1. Initialize the system:

Set up the required hardware, such as a camera or sensor, for hand tracking. Import the necessary libraries and dependencies, including OpenCV and MediaPipe. Configure the canvas parameters, such as size and resolution.

2. Start capturing video frames:

Initialize the video capture stream to obtain frames from the camera. Preprocess each frame, if required, for improved hand tracking performance. Convert the frame to the appropriate format for further processing.

3. Apply hand tracking:

Use the hand tracking model provided by MediaPipe to detect and track hand landmarks. Retrieve the hand landmarks, such as fingers, palm, and wrist, from the model's output. Perform any necessary transformations, such as scaling or normalization, on the landmark coordinates.

4. Map hand movements to the canvas:

Define the mapping between the hand movements and the canvas coordinates. Calculate the position of the hand on the canvas based on the detected landmarks. Update the position of the drawing tool, such as a virtual pen or brush, on the canvas accordingly.

5. Draw on the canvas:

Implement the drawing functionality, such as line or shape rendering, on the canvas. Detect the user's desired drawing action, such as pressing a button or making a specific gesture. Update the canvas based on the hand movements, allowing the user to draw in the air.

6. Display the canvas:

Render the canvas with the drawn content on the screen or display device. Continuously update the canvas as the user moves their hand in the air. Ensure a smooth and responsive visual feedback to provide a seamless drawing experience.

7. Handle user interactions:

Implement user interactions, such as clearing the canvas or changing the drawing tool. Define the gestures or commands recognized by the system to perform specific actions. Continuously monitor the user's input and respond accordingly.

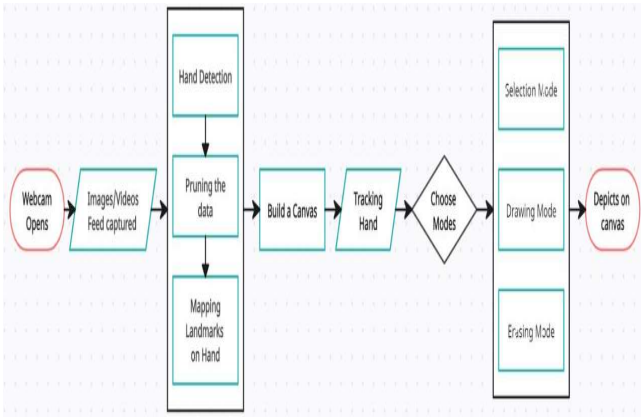

Fig 3: Application Workflow

VII. CONCLUSION

While Air Canvas demonstrates promising results, there are some limitations to consider. Challenging lighting conditions, occlusions, and background clutter may affect the accuracy of hand tracking. Additionally, the system's performance may vary depending on the hardware capabilities and computational resources available. Nevertheless, Air

Canvas opens up possibilities for interactive digital art, design, and other creative applications. With further optimizations and refinements, it has the potential to become a valuable tool for artists, designers, and individuals looking for innovative ways to interact with digital content. In conclusion, Air Canvas showcases the potential of using OpenCV and MediaPipe for hand-tracking applications. It's successful implementation and accurate tracking of hand movements provide a foundation for future advancements in this field.

VIII. Future Scope

Air Canvas has immense future potential as a platform for artistic expression, digital design, and interactive experiences. Its accurate hand-tracking capabilities using OpenCV and MediaPipe can be further enhanced to include gesture recognition, allowing users to perform specific actions by making predefined hand gestures. The system can be integrated with augmented reality (AR) and virtual reality (VR) technologies, enabling users to create immersive 3D artworks in a virtual space. Expanding the system's compatibility with various devices and operating systems would broaden its accessibility and user base. Ongoing research and development can focus on refining the system's performance under challenging conditions and exploring additional features to enhance the overall user experience.

Acknowledgement

We would like to thank our guide Dr. Mohammed Zabeeulla A N, Deptartment of Computer Science and Engineering, Jain (Deemed-to-be University), for sparing his valuable time to extend help in every step of our project work, which paved the way for smooth progress and the fruitful culmination of the project. We would like to thank all who directly or indirectly helped us complete the Project work successfully.

References

[1] K Sai Sumanth Reddy, Jahnavi S, Abhishek R, Abhinandan Heggde, Lakshmi Prashanth Reddy "Virtual Air Canvas Application using OpenCV and Numpy in Python" Volume 10, Issue 4 April 2022 IJCRT

[2] G.Vijaya Raj Siddarth, D. Vijendra Kumar, I.Vishnu Vardhan Reddy, R. Venkata Satya Sravani, Y. Lalitha Sri "Building a Air Canvas using Numpy and Opencv in Python" IJMTST, 8(S05):159-164, 2022.

[3] M. Bhargavi, Dr. B. Esther Sunanda, M.R.S. Ananya, M. Tulasi Sree, N. Kavya "Air Canvas Using Opencv, Mediapipe" IRJMETS Volume:04/Issue:05/May-2022.

[4] Zhou Ren, Junsong Yuan, Jingjing Meng, Zhengyou Zhang "Robust Hand Gesture Recognition with Kinect Sensor" 1520-9210/$31.00 © 2013 IEEE.

[5] Andrea Urru, Lizy Abraham, Niccolò Normani, Mariusz P. Wilk ID, Michael Walsh, Brendan O'Flynn "Hand Tracking and Gesture Recognition Using Lensless Smart Sensors" Sensors 2018, 18, 2834.

[6] Sahil Agrawal, Shravani Belgamwar "An Arduino based Gesture Control System for Human Interface" 978-1-5386-5257-2/18/$31.00 ©2018 IEEE.

[7] Revati Khade, Prajakta Vidhate, Saina Rasal "Virtual Paint Application By Hand Gesture Recognition System" International Journal of Technical Research and Applications e-ISSN: 2320-8163,Volume 7, Issue 3 (MARCH-APRIL 2019)

[8] S.U. Saoji, Akash Kumar Choudhary, Nishtha Dua, Bharat Phogat " Air Canvas Application Using OpenCV And Numpy in Python" IRJET Aug 2021 Volume: 08 Issue: 08

[9] P. Ramasamy, R. Srinivasan, G. Prabhu , "An economical air writing system is converting finger movement to text using a web camera" ICRTIT Chennai, pp. 1-6, 2016.

[10] Adinarayana Salina , K. Kaivalya , K. Sriharsha , K. Praveen , M. Nirosha " Creating Air Canvas Using Computer Vision " IJAEM Volume 4, Issue 6 June 2022.

[11] Aadesh , Hassan , Sahil , Dr H.S Guruprasad " Alphabet Detection through Air Canvas Using Deep Learning and OpenCV " 2022 IJRTI | Volume 7, Issue 7.

[12] Sayli More, Prachi Mhatre, Shruti Pakhare, Surekha Khot "Air Canvas: Draw in Air" IRJET Volume:09 Issue: 02 | Feb 2022.

[13] Dr. M. Parameswari, E. Vaishnavi, L. Priyanka, G. R. Swetha, T. Uma "Air Canvas Application Using Opencv And Numpy In Python" IJERME Volume 7, Issue 1, 2022.

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

[15] Manoj Kumar Rajagopal, Srinivas K, "Study of Hand Gesture Recognition and Classification" Asian Journal of Pharmaceutical and Clinical Research Special Issue(April),2017.