Creating Virtual Interactive Board Using Computer Vision

Pardeep Tyagi
KIET Group of Institutions
Ghaziabad, Uttar Pradesh
pardeep.tyagi@kiet.edu

Vaibhav Singh
KIET Group of Institutions
Ghaziabad, Uttar Pradesh
vaibhav.2024cs1090@kiet.edu

Vaibhav Mittal
KIET Group of Institutions
Ghaziabad, Uttar Pradesh
vaibhav.2024cs1007@kiet.edu

Sarthak Srivastava
KIET Group of Institutions
Ghaziabad, Uttar Pradesh
sarthak.2024cs1037@kiet.edu

ABSTRACT

One of the most intriguing and difficult developments in the fields of image processing and pattern design recently has been the ability to write in the air. It primarily advances an automated process and can enhance the interface between a machine and a human in a variety of applications. New strategies and procedures that help in cutting processing time and offering high recognition efficiency and accuracy have been the focus of effort in a number of research fields. In the realm of computer vision, object tracking is viewed as a crucial task. It entails first identifying the item, then following its motion from frame to frame, and then analysing the object's behaviour. We will follow the finger's movement using computer vision. It will be a potent technique of communication and a successful way to lessen the need for writing. As Everyone is aware that painters produce paintings on canvases. Consider the possibility that we could paint in the air merely by waving our hands. Therefore, in this project, we'll use Python and OpenCV to create an air canvas. Computer vision, hand and real-time gesture control, air writing, and object identification are some related terms.

I. INTRODUCTION

Modern literature is being supplanted by digital art as a result of the advent of the digital age. The expression and dissemination of an art form using a digital medium is referred to as "digital art." The term "traditional art" describes the types of art produced before digital art. It can be simply broken down into the visual arts, the audio arts, the audio- visual arts, and the audio-visual fantasy arts from the reception to the analysis. Traditional and digital art are related and rely on one another. The current condition of affairs includes a symbiotic relationship between conventional and digital art, so recognizing the Pen and paper, chalk, and the chalkboard writing technique are examples of traditional writing techniques. The creation of a hand gesture recognition system for digital writing is the fundamental aim of digital art. Digital art can be created using a variety of writing tools, including keyboards, touchscreens, digital pens, styluses, electronic gloves, and more. However, in our system, we combine hand gesture detection with computer vision and Python programming to produce a genuine human -machine interface. The necessity for developing natural human-computer interaction (HCI) solutions to replace existing systems is growing quickly as a result of technological improvements. Any colour within a specific range of the HSV colour space can be detected using the image processing technique known as colour detection. One method of labelling each pixel in an image, when each pixel shares attributes, is called image segmentation. In this project, we employed a 64 -bit operating system, an Intel(R) Core (TM) i5 10500H processor running at 4.0GHz, 16 GB of RAM, and Python, NumPy, and OpenCV software libraries

II. LITERATURE SURVEY

A. Strong hand recognition system using the Kinect sensor. The suggested technique in makes advantage of the Kinect sensor 's depth and colour data to determine the hand's shape. With the Kinect sensor, gesture recognition presents a challenging issue to be acknowledged. This particular Kinect sensor only has a 640x480 resolution. It is effective for tracking huge objects like the human body. However, even a finger 's size is difficult.

B. The movement of a finger with an LED. the authors have suggested a technique where An LED is mounted on the user's finger, and a web camera is used to monitor it. Characters from the plot and the database are compared with one another. The returning alphabets are all those that match the sketched pattern. It needs an aiming red LED light source that is installed on the finger. Assume that only LED light is in view on the web camera and that no other red objects are nearby.

C. Improved Desktop User Interface An improved segmented desktop interface concept for interactivity was put forth in [5]. You can use desktop apps with this system, which uses a projector and a charged device (CCD) camera that you can use with your fingertips. Each component of this system completes a task that is wholly distinct from the others. Radial menus are selected with the left hand, while manipulative objects are chosen with the right hand. His method involves the use of an infrared camera. The algorithm creates search windows for the fingertip because it is computationally difficult to identify the fingertip.

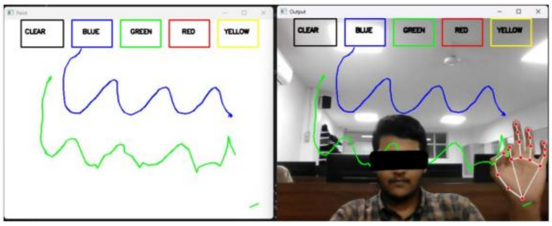

Figure 2.1 *Canvas Based Drawing*

2.1 CHALLENGES IDENTIFIED

A. The method used for fingertip identification only works with your fingers; highlighters and other such gadgets are not compatible. Without a sophisticated equipment like a depth sensor, it is extremely difficult to recognize and identify an object like a finger from an RGB image.

B. The technology uses an RGB camera to begin writing and does not allow for pen up and down movement. The inability to detect depth prevents tracking of the pen's up and down movement. As a result, the fingertip 's complete trajectory is traced, and the resulting image is useless and unrecognizable to the model. Figure illustrates the distinction between handwritten and air written doodle, write, draw, etc., we use a lot of paper. However, in this system, we employ hand gesture detection along with machine learning algorithms and Python programming to produce a genuine human-machine interface.

C. Our initiative primarily focuses on addressing these significant issues:

1. Deaf people: Sign language is a useful tool for communication even if we sometimes take hearing and listening for granted. Without a translator in the middle, the majority of people in the world are unable to grasp their emotions.

2. One of the primary focuses of our virtual interactive board is to reduce paper wastage. By providing an alternative to traditional paper-based methods, this technology can contribute to saving trees and reducing the environmental impact associated with paper production and disposal.

3. Accessibility: to a wide range of users, including those from underserved communities and regions with limited access to traditional educational resources.

D. Control of real -time systems It takes a lot of coding attention to change a system's status in real- time using hand gestures. In order to completely master his plan, the user must also be familiar with numerous motions.

2.2 PROBLEM STATEMENT

You probably had no idea that waving your finger in the air may aid in creating a realistic image. The fact that this aerial web functions in computer vision projects is fantastic.

2.3 PROBLEM SOLUTION

The evolution of numerous writing methods, including the usage of keyboards, touchscreen surfaces, digital pens, styluses, wearing electronic gloves, and more, is made possible by computer projects. These projects allow us to quickly draw on a screen by waving our fingers in the direction of the specified colour.

III. FEATURES

1. Has the ability to track any particular person's necessary colour pointer.

2. Users may easily and quickly draw with four distinct colours, and they can even switch them at any time.

3. To clean the board one piece at a time, choose the Clear option at the top of the display.

4. After the software has launched, there is no need to make contact with the computer.

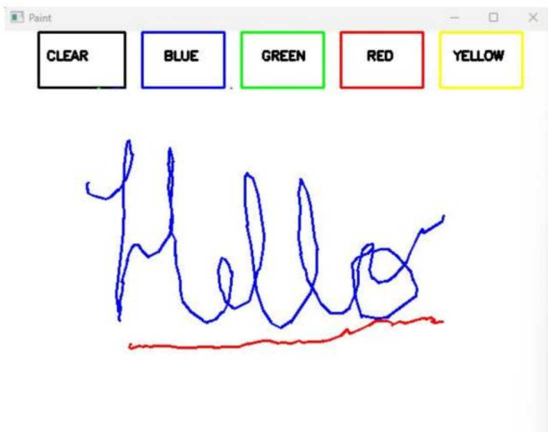

Figure 3.1 *UI of Virtual Interactive Board*

IV. SYSTEM METHODOLOGY

The finger detection model of the system demands a data collecting. The main goal of the fingertip model is that a single-color stylus or air pen can be used to write with the fingertip sensor model for air writing. The technique, however, makes use of fingertips. Without the need for a stylus, this recorded movement is captured through drawing characters that individuals can write in the air. To create a list of coordinates, we employed deep learning algorithms to find the fingertip in each image. Technique for creating finger recognition dataset:

A. Video to Frames: This technique captures hand motions in two-second snippets captured at different angles. Diagram 3 then shows how these movies are divided into 30 different pictures, for a total of 2000 images. This dataset was manually labelled using Labelling. The best model with this dataset has an accuracy of 99%. However, because all thirty of the frames were produced from the same movie and location, the dataset is incredibly repetitious. Consequently, the model exhibits poor performance for discrete backgrounds that differ from the backgrounds found in the data set.

B. Take pictures against different backgrounds: We added a new dataset to address the drawback of the earlier method's lack of diversity. This time, we are conscious of the fact that controlling the system requires a few motions. Consequently, we gathered four distinct hand positions, which are displayed in Figure 4.

C. Training the finger recognition model: The dataset is split into training and development groups (85% to 15%) after it is ready and labelled. Our dataset was trained using pre -trained faster Single Shot (SSD) and RCN detector models. When it comes to precision, Fast RCN outperforms SSD. Information needs to be verified, so we do that. Two common detection modules one that categorizes regions and the other that makes suggestions are combined in the SSD.

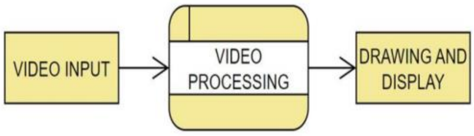

Figure 4.1 *Level 1 data flow diagram of the program's primary function*

The primary goal was to enable the model to distinguish between the four fingers' fingertips. As a result, the user can now control the system using the number of fingers that he represents. With the help of their index finger, they can now type rapidly. Two fingers can be used to convert that writing motion into electronic text, three fingers can add space, five fingers can be used to press the backspace key, four fingers can be used to make consecutive predictions, and three fingers can be used to select the first, second, or third prediction based on the requirement. Exit the prediction mode by displaying five fingers. There are roughly 1800 photos in this dataset. The prior model is intended to automatically label this dataset using this type of script. After that, we offer a new template and fix the incorrectly labelled photos. 94% accuracy is attained. Unlike the previous model, this one performs effectively in a variety of settings.

Figure 4.2 *Detection of Finger Tips Using OpenCV*

V. WORKFLOW ALGORITHM

The most intriguing feature of our system is this. Writing brings about the introduction of features. As a result, the quantity of gestures required to operate the system is equivalent to the quantity of actions. The fundamental components of our system are as follows:

1. Writing Mode: The system will draw and store the fingertip coordinates in this stage.

2. Colour Mode: The user can select from a variety of available colours for the text.

3. Backspace mode: In the event that a user makes a mistake, we need to fix it, hence we need a backspace. There is a new motion to add a backspace.

An explanation of the following steps will help you create an air canvas using OpenCV:

• Import and install the required files and packages

• Open a canvas window to draw on

• Capture frames from the camera

• Detect and track the green colour

• Draw on the canvas window.

VI. SYSTEM DESIGN

Ever wished you could just raise your finger in the air and catch your imagination? here, we'll use the webcam to record the movement of the colour marker to build an aerial canvas that can be used to sketch anything on it. A marker is an object of colour that is placed on the tip of the finger. In this project, computer vision techniques are aided by OpenCV. Though knowing the fundamentals can be used in any language that OpenCV supports, Python is the ideal choice due to its abundance of libraries and simple syntax. Here, colour detection and tracking are employed to accomplish our objective. Once a colour marker is found, a mask is created. It contains the subsequent phases of the product mask's morphological activity, including erosion and expansion.

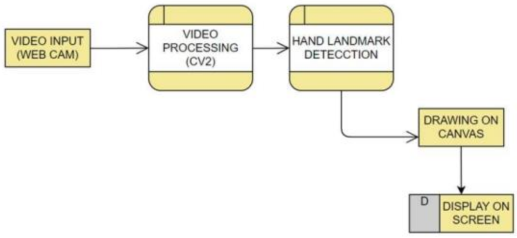

Figure 6.1 *Level 2 data flow diagram of the program*

A. Colour tracking of object at fingertips: To identify coloured objects, the webcam captured image must first be transformed to the HSV colour system. This code helps with colour space for colour tracking by converting the entering image to HSV space, which is fantastic. Now that we have the coloured item on our fingers, we will make track bars to align the HSV values into the necessary colour range. After track bars are configured, we obtain the track bars' current value and construct a range. A NumPy struct called range is used to transmit data into the cv2.inrange*()* function for additional processing. The coloured object and mask are returned by this function. This mask is incorporated in a black and white image that has white pixels where the necessary colour is needed.

B. Detecting the mask contour of the coloured object: It's time to locate the mask 's centre in order to draw the line after finding it in the air canvas. Here, we manipulate the mask morphologically in the code below to remove imperfections and make the outlines easier to see.

C. Drawing the Line by using the Contour position. The actual reasoning behind this Computer Vision project is that we'll execute a Python deque on a data structure. The contour's position on each frame will be stored in the deque, and we will use these points to create a line using OpenCV methods. This contour's location aids in our ability to decide whether to draw on the sheet or to press a button. Some of the buttons at the top of the canvas have been implemented; if the pointer lands in their area, it will assist in initiating their function. On the canvas made with OpenCV, there are four buttons. By dequeue, clear the entire screen. Red: Use the palette to change the ink's colour to red. Green: Use the palette to change the colour to green. Yellow: Make the hue yellow.

D. Points used: We will now draw all the points on the locations stored in the dequeue, with the corresponding colour.

6.1 PROJECT FLOW

1. Play back the captured image and convert it to the HSV colour space (which makes colour easy to identify).

2. Create the canvas and embellish it with the matching-coloured buttons.

3. Modify the trace bar values in order to identify the colour highlight mask.

4. Mask pre-treatment using morphological adjustments.

5. Locate the contours, note the coordinates of the largest contour's centre, and continue storing them in an array for the upcoming frames. (Tables for doing sketches on the painting)

6. Lastly, sketch the points that are kept in the table.

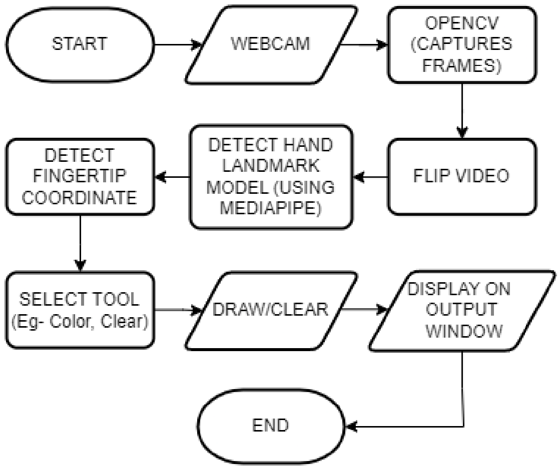

Figure 6.2 *Flow Chart representation*

VII. RESULT

These are our project's output and results. To run our function, we used the OpenCV module. We ran our code in anaconda, Jupiter Notebook, where we first installed the necessary packages before running the code to get the desired results. Two windows are visible here.

Our code specifies coordinates for our pen to be monitored on the camera page, which is the first window. We can utilize components like clear or colour change in the second window to depict the recorded coordinates.

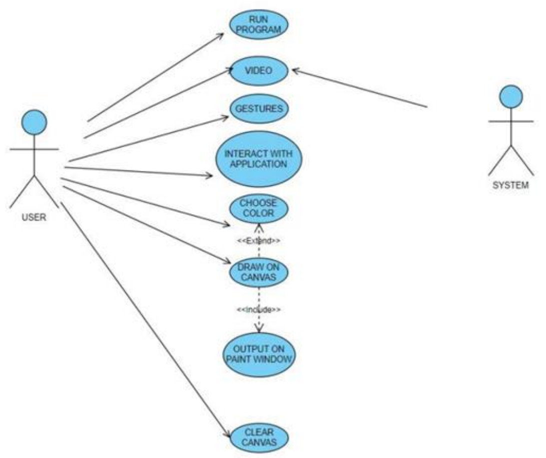

Figure 6.3 *Use Case diagram representation*

VIII. CONCLUSION

The system has the ability to undermine conventional writing techniques. It will be very helpful in facilitating communication for those with specific difficulties. The technology can be easily used by the elderly or people who have trouble using the keyboard. The technology can be extended to enable rapid control of Internet of Things devices. In the future, some system restrictions might be strengthened. First off, writing will be sped up by enabling word-for-word writing through the use of handwriting recognition rather than character recognition. Secondly, instead of using your fingers to manage the system in real time, you can utilize hand motions with pauses, as demonstrated in [1]. Third, our method modifies the initial state of fingertip recognition in its background state. It is imperative that the air writing system only responds to the control movements of its owner and not be tricked by people in close proximity. Future item detection algorithms, such as Yolo v3, will increase the precision and speed of fingertip recognition. Artificial intelligence developments in the future will make aerial writing more effective.

REFERENCES

[1] Huang, Yichao, Xiaorui Liu, Xin Zhang, and Lianwen Jin. "A pointing gesture based egocentric interaction system: Dataset, approach and application." In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 16-23. 2016.

[2] Mehtab, Sidra. "Object Detection and Tracking Using OpenCV in Python."

[3] Saoji, S., Nishtha Dua, Akash Kumar Choudhary, and Bharat Phogat. "Air canvas application using Opencv and numpy in python." IRJET 8, no. 08 (2021).

[4] Satpute, Suraj, Harshad Shende, Vikas Shukla, and Bharti Patil. "Real time object detection using deep-learning and OpenCV." International Research Journal of Engineering and Technology (IRJET) 7, no. 4 (2020): 3243-3246.

[5] Chang, Yuan-Hsiang, and Chen-Ming Chang. Automatic hand-pose trajectory tracking system using video sequences. IntechOpen, 2010.

[6] Gurav, Ruchi Manish, and Premanand K. Kadbe. "Real time finger tracking and contour detection for gesture recognition using OpenCV." In 2015 International Conference on Industrial Instrumentation and Control (ICIC), pp. 974-977. IEEE, 2015.

[7] Chandhan, Tamalampudi Hemaa, Nalin Raj, Neelam Nanda Kishore Reddy, and Mohammed Zabeeulla AN. "Air Canvas: Hand Tracking Using OpenCV and MediaPipe." In 1st-International Conference on Recent Innovations in Computing, Science & Technology. 2023.

[8] Tiwari, Satyam V., Deep A. Vartak, and Ms Soniya Khatu. "Virtual Hand Gesture Painting."

[9] Vasavi, R., Nenavath Rahul, A. Snigdha, K. Jeffery Moses, and S. Vishal Simha. "Painting with Hand Gestures using MediaPipe." Int. J. Innov. Sci. Res. Technol. 7, no. 12 (2022): 1285-1291.

[10] Khanum, Hajeera, and H. B. Pramod. "Smart Presentation Control by Hand Gestures Using computer vision and Google's Mediapipe." International Research Journal of Engineering and Technology (IRJET) (2022): 2657.

[11] Agrawal, Subhash Chand, Rajesh Kumar Tripathi, Neeraj Bhardwaj, and Prashun Parashar. "Virtual Drawing: An Air Paint Application." In 2023 2nd International Conference on Edge Computing and Applications (ICECAA), pp. 971-975. IEEE, 2023.

[12] KANTER, JORDAN A., and KAMIL QUINTEROS. "GESTURAL DESIGN-HAND TRACKING FOR DIGITAL DRAWING." Architecture and Planning Journal (APJ) 28, no. 3 (2023): 3.

[13] DAHANAYAKA, DTDM, AR LOKUGE, JADE JAYAKODY, and IU ATTHANAYAKE. "Smart Drawing for Online Teaching." Instrumentation 8, no. 2 (2021): 56-66.

[14] Gunda, Chidvika, Manohar Maddelabanda, and Hariharan Shanmugasundaram. "Free hand text displaying through hand gestures using mediapipe." In 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), pp. 996-1000. IEEE, 2022.

[15] Osama, NourEldin, Yousr Ahmed, Hussein Mohamed, Seif Eldin Hesham, Youssef Ahmed, Eman K. Elsayed, and Dalia Ezzat. "Virtual Control System for Presentations by Real-Time Hand Gesture Recognition Based on Machine Learning." In International Conference on Advanced Intelligent Systems and Informatics, pp. 327-335. Cham: Springer Nature Switzerland, 2023.