AirSketch: Virtual Hand-Drawing

Mrudani Hada
Electronics & Communication Engineering Department
Institute of Technology, Nirma University, Ahmedabad, India
20BEC071@nirmauni.ac.in

Parth Parikh
Electronics & Communication Engineering Department
Institute of Technology, Nirma University, Ahmedabad, India
20BEC081@nirmauni.ac.in

***Abstract*— Air Sketching has become one of the latest areas of research as it is the most fascinating application of image processing, pattern recognition and computer vision all put under a single roof. Progressions in object tracking has been possible due to inventions of fast processing computers and multiple inexpensive good quality video cameras. At the same time, it invites myriads of difficulties like determination of suitable article representation, feature selection for tracking, object recognition and object tracking. In this paper, we have implemented an algorithm which can track the path sketched by a significantly distinct and detectable object and mirrors it on a white canvas to digitize it. This coursework has various applications in the digital world namely e-signatures, air-to-text conversions, motion detection and tracking, fun activity workspace for children, etc.**

Keywords—Air sketching, object detection, object tracking, computer vision, character recognition, real-time gesture control, numpy, collections

I. Introduction

As the world does its best to move on from the pandemic, one of the lasting legacies for many advanced economies has been greater adoption of digital technologies. Using a PC and a web camera, when we involve the OpenCV library of python language for object detection, here, a blue coloured distinctly detectable object, can reproduce whatever is sketched in the air by the user to a blank white canvas. It transfers hand signals onto the screen. Particularly, we use colour detection to accomplish our task. This form of an arrangement can prove to be very useful and effective manner of communication for the deaf and dumb. Teaching and educational fields are the maximum profiters out of such a scenario. Digitization of traditional writing on pen and paper has so many applications like motion-to-text conversion, an aid for the deaf and dumb community to communicate and express their feelings with the outer world without a translator, have an active participation in saving the environment and preventing carbon footprint in the paper industry by savings thousands of tree from being cut for paper production, replacing conventional blackboards and enhancing the teaching-learning process.
Our main focus is in using the python language due to vast bank of libraries that the coding language possess. The OpenCV library, numpy library and the collections library are the highlights of this paper.

Libraries Used

A. OpenCV (Open Source Computer Vision Library)

OpenCV is a powerful open-source computer vision and machine learning software library that provides various tools and functions to perform object detection. Object detection is the process of locating and classifying objects within an image or video stream. OpenCV is commonly used for this task due to its extensive set of features and algorithms.It offers the following Object Detection Algorithms:

1. Haar Cascades: These are trained XML files that can detect specific objects or features in images.

2. HOG (Histogram of Oriented Gradients): This method detects objects based on the distribution of gradient orientations in an image.

3. YOLO (You Only Look Once): A deep learning-based algorithm that is capable of real-time object detection.

4. SSD (Single Shot Multi-Box Detector): Another deep learning-based method for object detection.

B. Numpy (Numerical Python) Library

NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy primarily deals with arrays. You can create arrays using various methods and constructors:

1. Creating an Array from a List
2. Creating an Array of Zeros or Ones
3. Creating an Array with a Range of Values
4. Creating an Array with a Specified Shape
5. Creating an Identity Matrix
6. Creating a Random Array

C. collections Library

The Python `collections` module is part of the Python Standard Library and provides a collection of specialized data structures that are not part of the built-in types like lists, dictionaries, and tuples. These data structures are highly optimized and designed to solve specific problems efficiently. Here are some of the main data structures provided by the `collections` module:

1. Counter: ‘collections.counter’ is a dictionary subclass for counting hash-able objects. It is commonly used for counting the occurrences of elements in an iterable loop.

2. DefaultDict: `collections.defaultdict is a dictionary subclass that returns a default value when a key does not exist in the dictionary.

3. OrderedDict: `collections.OrderedDict` is a dictionary subclass that maintains the order of items based on their insertion order.

4. NamedTuple: `collections.namedtuple` is a factory function for creating tuple subclasses with named fields. It's a convenient way to define simple classes for storing data.

5. Deque: `collections.deque` is a double-ended queue that allows efficient and fast append and pop operations from both ends. It is a useful data structure for implementing queues and stacks.

6. ChainMap: `collections.ChainMap` is a class for creating a single view of multiple dictionaries. It encapsulates multiple dictionaries into one and allows you to search them as if they were a single dictionary.

II. Literature Riview

A. Tracking of Brush Tip on Real Canvas : Silhoutte-Based and Deep Ensemble Network-Based Approaches

Author-Joolekha Bibi Joolee , Ahsan Raza, Muhammad Abdullah, Seokhee Jeon

Working- The proposed advanced painting system operates by utilizing data collected through an external tracking device (Optitrack V120) and a design-based approach. During the actual painting process, the system assesses the position of the brush tip by treating the brush handle as data, enabling the use of real materials with a physical brush. Throughout the testing phase, the system operates continuously, tracking the current position (both location and orientation) of the brush handle. The advanced painting system then takes this brush handle data and predicts the brush tip's real-time position. Data collection involves performing various brush strokes on a surface for 60 seconds in a controlled environment.

B. An Implementation of virtual white board using Open CV for virtual Classes.

Author- S.V. Aswin Kumer⇑, P. Kanakaraja, Sheik Areez, Yamini Patnaik, Pamarthi Tarun Kumar

Working- In this paper, the authors describe their use of OpenCV for object detection, enabling users to write in the air. They implemented this using the Python programming language due to its simplicity and ease of understanding. When the code is executed, it displays a white screen on the monitor, and any writing performed in front of the camera is tracked based on predefined properties in the code. These tracked points are then displayed on the screen.

C. AIR CANVAS APPLICATION USING OPENCV AND NUMPY IN PYTHON

Authors- Saoji , Saurabh & Dua, & Vidyapeeth, Bharati & Choudhary, Akash & Phogat, Bharat

Working-The project's main objective is to convert motion into art. The system makes use of a camera device and computer vision software to track the movement of our fingertip. This innovative technical approach minimizes the use of mobile and PC devices by eliminating the necessity for manual writing.

D. Virtual Ink Using Python

Authors: Mishra, P., & Uniyal, A.

Working- The project relies on Python and utilizes two key libraries, OpenCV and NumPy. OpenCV is a library primarily designed for computer vision, initially developed by Intel and later supported by Willow Garage. On the other hand, NumPy is a Python library that enhances support for large, multi-dimensional arrays and matrices, offering a wide range of high-level numerical functions to manipulate these arrays. The project primarily functions as a computer vision application that leverages your device's webcam. To use it, simply hold a pencil or pen in your hand and move it in the air in front of your device's camera, typically within a range of about 15-30 centimeters.

E. VIRTUAL SKETCH USING OPENCV

Authors- Srungavarapu, Pranavi & Maganti, Eswar & Sakhamuri, Srilekkha & Veerada, Sai & Chinta, Anuradha.

Working-The papers focus is on performing morphological operations, which encompass a set of tasks for processing images based on their shapes. These operations involve applying a structuring element to an input image to generate an output image. The project involves establishing a connection between the human hand and the system through the utilization of OpenCV techniques and the Python programming language. This connection allows the user to select colors and draw on a designated area using their hand.

F. 3D Drawing with Augmented Reality

Author-Sharanya M, Sucheta Kolur , Sowmyashree B V, Sharadhi L, Bhanushree

Working- An Android mobile app enables users to draw on their surroundings as if it were a canvas. It offers real-time synchronization of drawings across all instances of the app in the same network space. Furthermore, it provides a tool for creative individuals to quickly sketch their ideas in 3D environments. The Freehand feature allows users to draw freely by moving their hand, and to begin drawing a line, they perform an air-tap gesture. The line is drawn continuously at the location of the cursor until the user ends it with another air-tap.

III. Challenges Encountered

A. Fingertip detection

The existing system only works with your fingers, and there are no highlighters, paints, or relatives. Identifying and characterizing an object such as a finger from an RGB image without a depth sensor is a great challenge.

B. Lack of pen up and pen down motion

The system uses a single RGB camera to write from above. Since depth sensing is not possible, up and down pen movements cannot be followed.

Therefore, the fingertip's entire trajectory is traced, and the resulting image would be absurd and not recognized by the model. The difference between hand written and air written 'G' is shown in Figure 1.

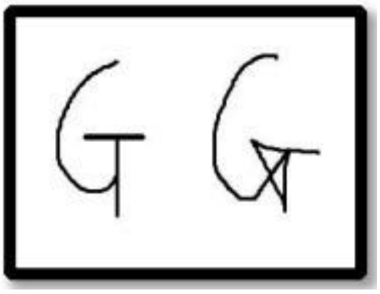

Fig 2. Difference between hand written and air written

IV. Block Diagram

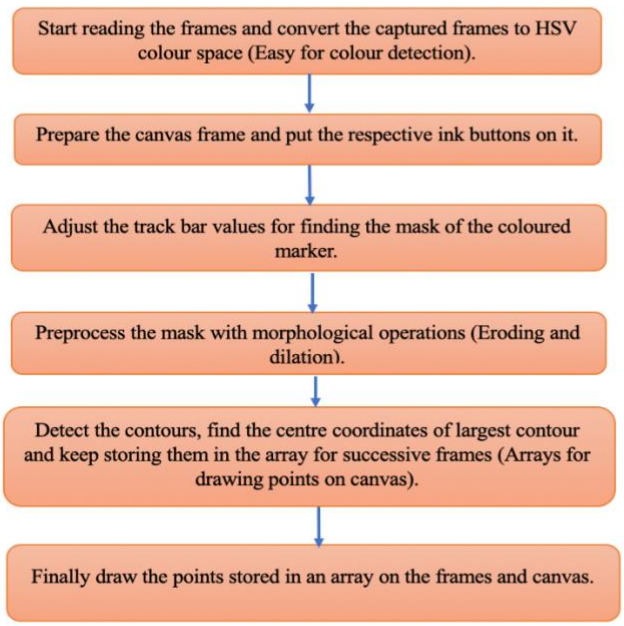

Fig 2. Flow of implementation

Here, the flow of the code is illustrated where step-wise processes are being followed. The camera will capture real-time video frames and then choose a detection color code, convert it to HSV. Prepare a white blank canvas board. Create a mask for the set color values and preprocess it by eroding and dilation. Prepare arrays for drawing points and replicate them on the canvas and store it in the form of array on the frames.

V. Implementation of Python Code

Algorithm-1 <u>Air Sketch</u>

1. Import libraries namely numpy, OpenCV and collections
2. Make a trackbar function considering the features of marker colour, upper and lower ranges
3. Start reading the frames and convert the captured frames to HSV color space (Easy for color detection).
4. Mention the indices used to mark the position of pointers in color arrays
5. Use the kernel for dilation purpose
6. Mention different ink colors such as blue, green, red and yellow
7. Setup the canvas and put respective ink buttons and clear all button
8. Load the default webcam of the PC
9. While True:
10. Read the frame from camera and flip it to your side. Furthermore, Adjust the track bar values for finding the mask of the colored marker.
11. Put text of all buttons with mentioned font size and colors
12. Preprocess the mask with morphological operations (Eroding and dilation).
13. If contours are formed:
14. find the center coordinates of largest contour and keep storing them in the array for successive frames (Arrays for drawing points on canvas).
15. Put the coordinates from the center so as to let the user click on each virtual button
16. Finally draw the points stored in an array on the frames and canvas.
17. **end**
18. If the user presses q to quit and close all the cameras
19. **end**
20. **end**

Firstly, import necessary libraries namely numpy, OpenCV and collections. There are four windows displayed on the screen as the code is implemented to be specific the paint window, the virtual screen on which the user will draw, the color detector window and the mask window.
Starting with the color detector window, we have used cv2.createTrackbar() so as to change the color adjustments. This is done as the selected color does not stick to a certain color, whereas it can be a shade of the color. Furthermore, run an infinite loop so as to detect the needed color and then break it as soon as the color is observed. Now run a loop if the above mentioned holds true such that the RGB colors are converted

to their respective HSV (hue, saturation, value) tones. Moreover, note the positions by cv2.getTrackbarPos() so as to set the upper_hue, upper_saturation, upper_value, lower_hue and lower_value. In our code the range of blue is specified. Here hue demonstrates is the spectrum containing all visible colors having a maximum of 180, the saturation is the purity and intensity of a color as displayed in an image. The higher the saturation of a color, the more vivid and intense it is. The lower a color's saturation, or chroma, the closer it is to pure gray on the grayscale and the color value refers to the relative lightness or darkness of a color thus both having a maximum value of 255.

Moving on to creating a mask, it displays a point on a black screen by cv2.inrange() to detect the position of the blue color contour. As this contains impurities, so to make it immune to impurities we need a kernel with an array of all ones, thus dilating the errors.

Additionally, the virtual screen is displayed which plays the role of the heat of the project. The script is written so as to decide the locations of the rectangles: CLEAR ALL, BLUE, GREEN, RED and YELLOW on the virtual screen. A function named cv2.putText() is used to put the determine the font, font size in those boxes displayed on the live frame. Furtherly, break the contour and find its length so as to detect the impurities. If its length is greater then zero then a yellow circle with specified radius and positions is illuminated on the detected object letting the user know that the writing is traced by the yellow circle.

Now you might have a question how the air-sketch work. A code is written for each successive array, where a dequeue is generated which is a type of array were removing and inserting values take place from both the sides. This dequeue are created for every color and are used to store the coordinates on which the contour is moving. After storing coordinates the pixels are stored in respective colors. Lastly, run a loop for each dequeue taking all coordinates associated with that color and imprinting that color on the screen. The clear all button resets all the dequeues.

To display the contents on the paint screen, create a paint window with specified dimensions and add 255 so as to showcase it as a sheet. Finally, show all the screens by using cv2.imshow() and then use the q on the keyboard to destroy all the windows created.

VI. Results

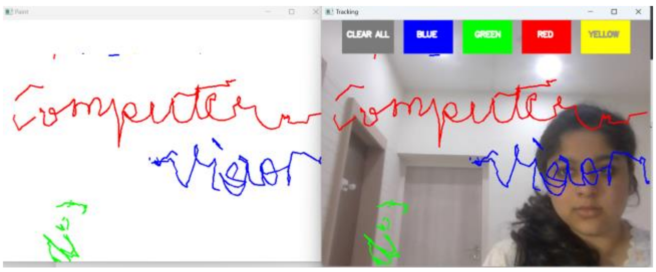

Fig 3. Demonstration of motion to text for colors red and blue

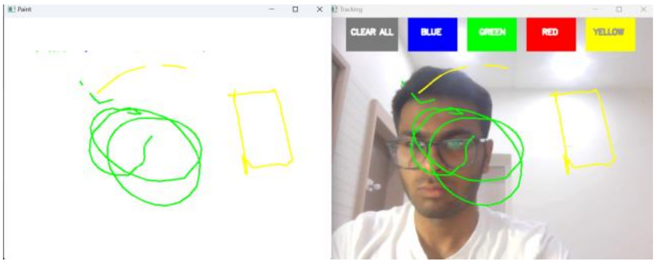

Fig 4. Demonstration of motion to shapes for colors yellow and green

VII. Conclusion

In this paper, we have implemented an air sketching algorithm using python as the programming language. Supporting libraries namely OpenCV, numpy and collections were used to fulfil the task. This arrangement will be crucial in challenging the conventional writing techniques. The user is given discretion to write in the air which will be masked and produced onto a white canvas board. This digitization of hand-written sketch to text is fascinating as well as has maximum utility. The model works efficiently and can be used for blue, green, yellow and red colors. A clear canvas taskbar is also introduced to clear the screen. Moreover, it follows and tracks the object whose color coded hues and saturation values are mentioned in the code. It can also track human finger when colored with detectable color. All in all, it is one of the finest applications of computer vision.

VIII. Future Scope

The realm of Computer Vision involves the scientific field dedicated to aiding computers in perceiving and interpreting digital images, such as photographs and videos. Over the course of several decades, it has been a subject of extensive research. Notably, computer vision has advanced to the point where it surpasses the pattern recognition abilities of the human visual system. In the healthcare sector, computer vision technologies have even outperformed human doctors in this aspect.

There are multiple factors to consider as computer vision continues to impact the human world. With further research and refinement, it is expected that computer vision will become more capable in the future. The training of these systems will become more straightforward, enabling them to identify more details in images than they can currently.

Computer vision is poised to collaborate with other technologies and subsets of artificial intelligence to create more sophisticated applications. For instance, in the case of image captioning applications, natural language generation (NLG) can be integrated to describe the surroundings for visually impaired individuals. Moreover, computer vision holds the potential to contribute to the development of artificial general intelligence (AGI) and artificial superintelligence (ASI) by processing information more effectively than the human visual system.

Furthermore, computer vision is an expanding field closely associated with virtual and augmented reality (VR and AR). Recent market participants have demonstrated a strong interest in the fusion of VR and AR, leading to a surge in the development and release of cutting-edge technological products in this domain.

REFERENCES

[1] Joolekha Bibi Joolee , Ahsan Raza, Muhammad Abdullah, Seokhee Jeon. " Tracking of Brush Tip on Real Canvas : Silhoutte-Based and Deep Ensemble Network-Based Approaches"

[2] S.V. Aswin Kumer⇑, P. Kanakaraja, Sheik Areez, Yamini Patnaik, Pamarthi Tarun Kumar, " An implementation of virtual white board using open CV for virtual classes.

[3] Saoji , Saurabh & Dua, & Vidyapeeth, Bharati & Choudhary, Akash & Phogat, Bharat. (2021). AIR CANVAS APPLICATION USING OPENCV AND NUMPY IN PYTHON. International Journal of Research in Engineering and Technology. 8. 2395-0056.

[4] Mishra, P., & Uniyal, A. (2021). Virtual Ink Using Python (No. 5707). Easy Chair.

[5] Srungavarapu, Pranavi & Maganti, Eswar & Sakhamuri, Srilekkha & Veerada, Sai & Chinta, Anuradha. (2021). Virtual Sketch using Open CV. International Journal of Innovative Technology and Exploring Engineering. 10. 107-108. 10.35940/ijitee.H9262.0610821.

[6] Sharanya M, Sucheta Kolur , Sowmyashree B V, Sharadhi L, Bhanushree. "3D Drawing with Augmented Reality"

[7] https://www.geeksforgeeks.org/create-air-canvas-using-python-opencv/ - accessed on 30/10/23

[8] https://www.youtube.com/watch?v=v4msPg1AdMY – accessed on 30/10/23

[9] http://www.workwithcolor.com/blue-color-hue-range-01.htm - accesed on 31/10/23

[10] http://www.workwithcolor.com/pink-color-hue-range-01.htm - accesed on 31/10/23