ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 11, November 2022

AI Virtual Painter using OpenCV and Mediapipe

[1] Dr. S. Nandagopal, [2] Soundarya R, [3] Vaishnavi S, [4] Vanisri S, [5] Hiranya S

[1] Principal of Nandha College of Technology, Erode, Tamilnadu, India
[2][3][4][5] Department of Computer Science and Engineering, Nandha College of Technology, Erode, Tamilnadu, India

Abstract*— Recognition of hand gestures has great importance for Human-computer interaction (HCI). The human hand is very small with complex junctions compared with the entire human body, hence recognizing the human hand is not an easy task. By using the hand gesture recognition the hand point/coordinates of hands can be detected using which we can make many impossible happens. Our work indicates one such finding, that is, VIRTUAL PAINTER. In our project, the main objective is to display the words on the monitor screen which we write on air in front of the webcam. This is done by recognizing the human hand through the normal webcam of a computer and the hand points are detected using the MediaPipe python library. Using the detected hand points the count of opened fingers is stored. When the index and middle fingers are open it means it is in Selection mode and when only the index finger is open it is in drawing mode. In Selection mode, the colors using which we want to draw can be selected from the list of colors that are made displayed on the screen. In drawing mode, the content that is written in front of the camera on air is drawn on the monitor screen. The application of this implementation can be used in many places where immediate implementation or explanation is needed.

***Keywords*— Air Writing, Character Recognition, Real-Time Gesture Control System, Computer Vision**

I. INTRODUCTION

The digital world is full of arts and writings. The significance of modern art paves a path for the modern world. This modern way of living is derived from the traditional cultures which were followed by our ancestors. In the case of writing, the traditional art contains multiple ways starting from manuscripts to paper and blackboard writing everything is a traditional way to express our thoughts in a written format. In this modern era, everything is getting digitized. So, the writing also must be enhanced digitally. Have anyone dreamt of air writing? Using modern technology that is also possible. All the contents that are written on air can be written on the screen without using any tools. The enhancement of this will make a new generation of digitized writing.

II. LITERATURE REVIEW

Artificial Intelligence is a technology in which the machine thinks like a human. The application of Artificial Intelligence plays a vital role in this modern era. The intelligence of humans is infused into the computer to gain more efficiency.

The Artificial Intelligence architecture is built by using some built-in technologies like OpenCV and MediaPipe. These are two open-source python libraries that are used to implement and manipulate real-life data to attain the desired outcome.

In [2015], the Author objects to developing a system that recognizes the hand gestures that can be used for controlling the robots or to communicate with others.

In [2015], the Author establishes an algorithm completely based on image processing using which the hand gestures are recognized to perform all the mouse and cursor movements using the identified hand gestures without touching the mouse.

In [2021], the Author aims to build a dataset using a single video. This is done by splitting the video into several images and all those images are processed to train the model. The only drawback is that this is not more accurate.

In [2016], the Author establishes a procedure by using LED which is placed on the finger. Through the webcam, the finger is tracked. The tracked finger is compared with the characters stored. The corresponding match is provided as the output.

III. EXISTING SYSTEM

Artificial Intelligence evolves in ample real-world applications in this digital era. Those applications pave a way for the Man-Machine interaction where all is about digital interaction rather than manual. The existing system is the one where the gesture of hand is recognized to help the deaf and dumb people to understand the hand symbols with ease virtually. In this, the gesture of hands is recognized and converted into characters that the hands symbolize. Using machine learning algorithms the hand gestures are extracted and represented as corresponding characters.

3.1 DRAWBACKS

- Low accuracy
- More time consuming
- less real-life application

IV. PROPOSED SYSTEM

In the proposed system, the hand coordinates are extracted through a webcam. These extracted hand points are processed and it is used to select the colors and to draw the contents on the monitor. Here the screen is considered as hundreds and thousands of (x, y) coordinates. The screen is split into several portions using these coordinates. Only using those portions the color selection split up is functioned and the rest

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 11, November 2022

of those portions are left for drawing. A list is maintained for all the color options to store the drawn contents on the monitor. Once the list is cleared the contents on the screen get erased. This application can be integrated with all the fields where the Virtual Painter is needed. It is a very useful application.

4.1 ADVANTAGES

- Reduce processing time.
- High recognition accuracy
- Easy to interact using virtual painting
- Advanced real-life application

V. MODULES & IMPLEMENTATION

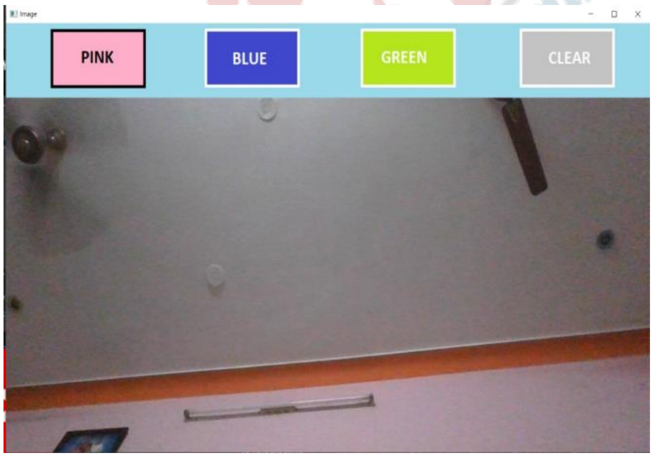
MODULE 1: SET THE FRAME

The first and foremost action is to set the screen with all the requirements. First, the list of colors that we prefer for drawing should be decided and then those colors must be added to the topmost section on the screen. If the user's hand moves above a threshold limit then the user could select the color and hand movement below the threshold would be for drawing the last selected color. Initially, the default color is assigned. Above the threshold limit, the screen is split into several boundaries of colors. Each boundary is noted and the corresponding color can be selected when the hand is inside the respective boundary.

SOURCE

```
cap = cv2. VideoCapture(0)
cap.set(3, 1280)
   cap.set(4, 720)
   while True:
      success, Image = cap.read()
      Image[0:125, 0:1280] = TopSection
      cv2.imshow("Image", Image)
      cv2.waitkey(1)
```

RESULT



MODULE 2: RECOGNIZE HAND POINTS

The next step is to recognize the hand points through the webcam. This is done by using the Mediapipe Python library which contains several built-in functions to proceed with the desired operations.

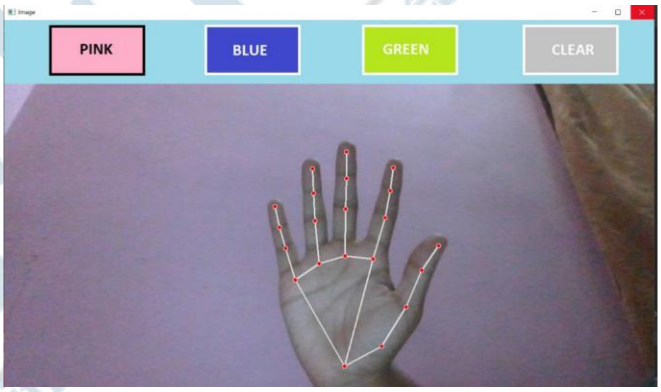
SOURCE

```
if output.multi_hand_landmarks:
    for i in output.multi_hand_landmarks:
       if d:

mpD.d_landmarks(image,i,mpH.HAND_CONNECTIONS)

if output.multi_hand_landmarks:
    myH = output.multi_hand_landmarks[number]
    for identity, landmark in enumerate(myH.landmark):
       height, weight, circumference = imgage.shape
       cx, cy = int(landmark.x * width), int(landmark.y
*height)
```

RESULT



MODULE 3: POSITION THE POINTERS

Before drawing we need to point out the pointer on the finger which is used for selecting and drawing the contents on the screen. There are two types of pointer that need to be indicated, one is for selection mode and the other is for drawing mode.

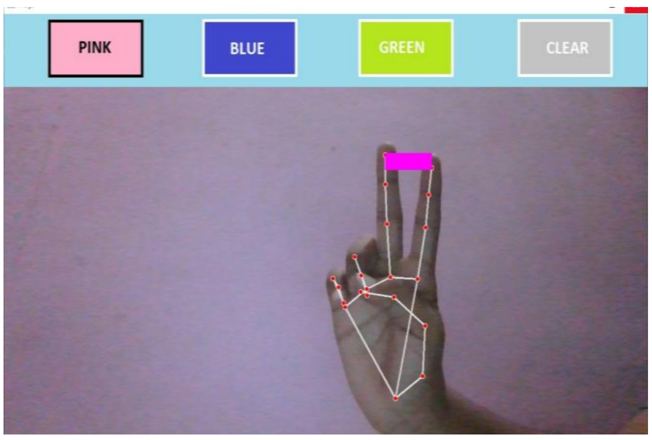
SELECTION MODE

Selection mode is the mode where we will select the desired color from the list of colors that appears on the screen. The selection mode is stimulated when the index finger and the middle finger is up. Once the selection mode is stimulated the marker will be indicated as a rectangle between those two fingers. This indicates to the user that it is in the selection mode. A default color and size are given to the rectangle.

SOURCE

```
if fing[1] and fing[2]:
cv2.rectangle(image, (x1, y1-25), (x2, y2+25), (255, 0,
255), cv2.FILLED)
```

RESULT



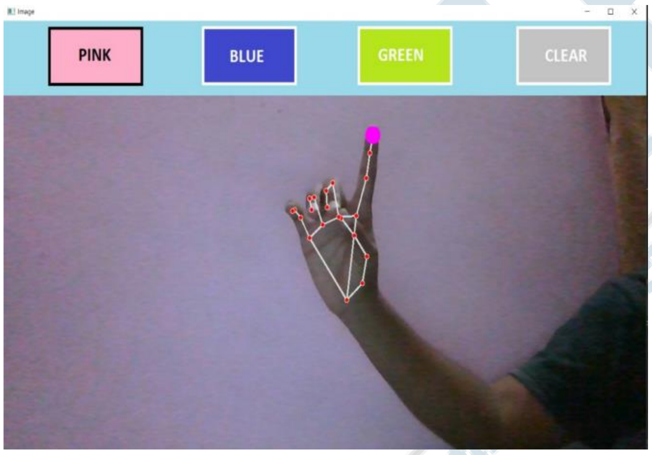
DRAWING MODE

Drawing mode is where we draw or write the thing which we want to showcase on the screen. This mode is stimulated when the index finger is up. This is indicated by a circle around the index finger. A default color and size are given to the circle.

SOURCE

```
if fing[1] and fing[2] == False:
cv2.circle(image, (x1, y1), 25, (255, 0, 255), cv2.FILLED)
```

RESULT



MODULE 4: SELECT COLORS

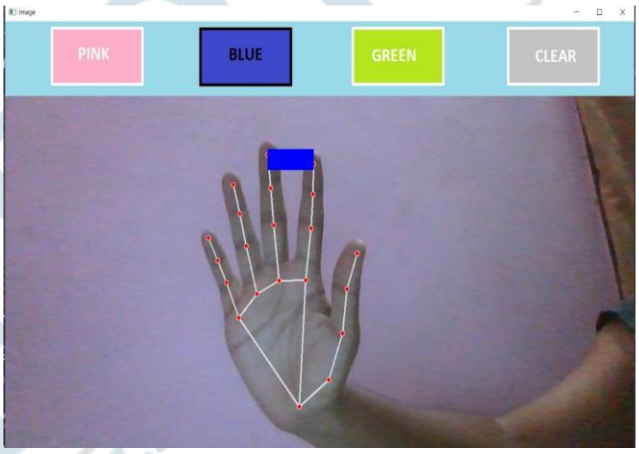
Once the frame is set, the real action starts. Now the color panel needs to be set on the top of the screen. This is done by separating the screen into several sections or boundaries. First, the frame must be split into five sections using the coordinates. For each section, a specific color is set. When **a** user moves the finger in each of the boundaries the respective color is selected.

SOURCE

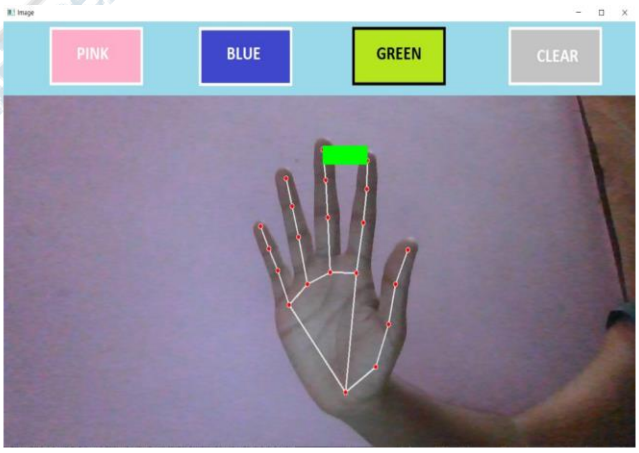
```
if y1 < 125:
    if 250 < x1 < 450:
        TopSection = color[0]
        draw = (255, 0, 255)
    elif 550 < x1 < 750:
        TopSection = color[1]
        draw = (255, 0, 0)
    elif 800 < x1 < 950:
        TopSection = color[2]
        draw = (0, 255, 0)
    elif 1050 < x1 < 1280:
        TopSection = color[3]
        draw = (0, 0, 0)
cv2.rectangle(image, (x1, y1-25), (x2, y2 + 25), draw, cv2.FILLED)
```

All the above values change as per the screen's configuration.

RESULT



COLOR 1



COLOR 2

MODULE 5: StART PAINTING

This is the place for what the whole process is made for. Now whatever the user writes on the air in front of the webcam will be shown on the screen. This is done by using the OpenCV python library and its built-in functions

ISSN (Online) 2394-2320

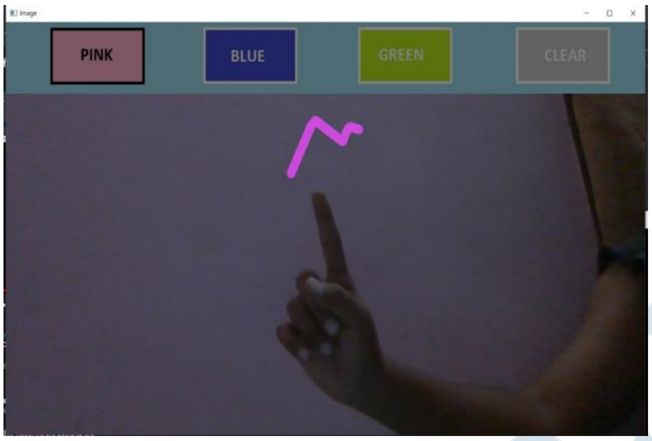
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 9, Issue 11, November 2022

SOURCE

```
if xprev == 0 and yprev == 1:
    xprev, yprev = x1, y1
cv2.line(image, (xprev, yprev), (x1, y1), Draw)
cv2.line(imagecanvas, (xprev, yprev), (x1, y1), Draw)
xprev, yprev = x1,y1
image = cv2.addweighted(image, 1, imagecanvas, 1, 0)
cv2.imshow("Image", image)
cv2.imshow("Canvas", imagecanvas)
```

RESULT



VI. CONCLUSION

The work can be integrated with any field where there is a need for a virtual painter. The framework can challenge customary composition/educating strategies. Framework usefulness alluded to the arrangement of capacities or administrations that the framework prepares for the users while the framework can work the explicit user needs movement proficiently like virtual drawing. This framework could be utilized as an option for showing programming utilized by educators. If further deciphered different virtual-based actual games could be made. The future enhancement can include more features that will help professional artists so that there will not be any more separate tools to be used for drawing, editing, sharing, and so on.

REFERENCES

[1] Y. Huang, X. Liu, X. Zhang, and L. Jin, "A Pointing Gesture Based Egocentric Interaction System: Dataset, Approach, and Application," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, pp. 370-377, 2016.

[2] P. Ramasamy, G. Prabhu, and R. Srinivasan, "An economical air writing system is converting finger movements to text using a web camera," 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, pp. 1-6, 2016.

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

[4] Alper Yilmaz, Omar Javed, Mubarak Shah, "Object Tracking: A Survey", ACM Computer Survey. Vol. 38, Issue. 4, Article 13, Pp. 1-45, 2006.

[5] Pranavi Srungavarapu, Eswar Pavan Maganti, Srilekkha Sakhamuri, Sai PavanKalyan Veerada, Anuradha Chinta, VirtualSketch using Open CV, ISSN: 2278- 3075 (Online), Volume-10 Issue-8, June 2021

[6] Pranavi Srungavarapu, Eswar Pavan Maganti, Srilekkha Sakhamuri, Sai PavanKalyan Veerada, Anuradha Chinta, VirtualSketch using Open CV, ISSN: 2278- 3075 (Online), Volume-10 Issue-8, June 2021

[7] First International Conference on Computer Engineering International Journal of Scientific Research in Science and Technology Virtual Painting with Opencv Using Python Yash Patil1, Mihir Paun1, Deep Paun1, Karunesh Singh1, Vishal Kisan Borate Volume 5 Issue 8, November-December-2020.

[8] Suryansh Pratap Singh a, Akshat Mittal b, Manas Gupta c, Soumalya Ghosh d, Anupam Lakhanpale Comparing Various Tracking Algorithms In OpenCV Vol.12 No.6 (2021), 5193-5198

[9] Bhumika Nandwana, Satyanarayan Tazi, Sheifalee Trivedi Dinesh Kumar khunteta, Santosh Kumar vipparthi, A Survey Paper on Hand Gesture Recognition

[10] Hemalatha Vadlamudi, Evaluation of Object Tracking System using Open-CV In Python International Journal of Engineering Research & Technology (IJERT)Vol. 9 Issue 09, September-2020

[11] Siddharth Mandgi1, Shubham Ghatge1, Mangesh Khairnar1, Kunal Gurnani1, Prof. Amit Hatekar2 Object Detection And Tracking Using Image Processing Vol. 8, Issue 2, (Part -1) February 2018, pp.39-41

[12] S. Soo 2014 Object detection using Haar-cascade Classifier (Inst. Comput. Sci. Univ. Tartu) vol. 2 no. 3 pp 1–12

[13] M. Nahar and M. S. Ali 2014 An Improved Approach for Digital Image Edge Detection (Int. J. Recent Dev. Eng. Technology) vol. 2 no. 3.

[14] It sees 2017 Open Source Computer Vision Library itseez2015opencv (Online) Available: https://github.com/itseez/opencv (Accessed: 13-Oct-2017)