Plagiarism - Report

Originality Assessment

10%

Overall Similarity

Date: Dec 8, 2023
Matches: 181 / 1765 words
Sources: 10

Remarks: Low similarity detected, check with your supervisor if changes are required.

Verify Report:

VIRTUAL AIR CANVAS USING OPENCV AND MEDIAPIPE

Dr.S.Subashini1, Pravin K.M2, Praveen Kumar K.J3, Pradheep K.K4, Sundharesh K.S5

1Asso. Prof, Dept. of Computer Science and Engineering, K.L.N. College of Engineering, Sivagangai,Tamilnadu,India

2,3,4,5 B.E. Final Year Student, Dept. of Computer Science,K.L.N. College of Engineering, Sivagangai,Tamilnadu,India.

Abstract

Recent development in artificial intelligence enables a digital environment to be navigated through human gestures. Particularly relevant in computer language instruction, gestures serve as nonverbal cues. The Virtual Canvas application transforms hand and finger gestures captured via webcam into a drawing tool. Leveraging OpenCV for robust image and video analysis and MediaPipe for streamlined perceptual computing, the project simplifies the development of applications involving tasks like hand tracking and face detection. Python, chosen for its ease of use and broad compatibility, serves as the project's language. By monitoring hand landmarks and finger IDs in real-time, the project enables dynamic modes such as selection, drawing, and canvas clearing. This research showcases the practical application of computer vision and gesture-based interactions, enhancing the user experience.

Keywords: OpenCV, MediaPipe, Hand Gestures, Hand Tracking, Computer Vision.

I. INTRODUCTION

The convergence of advanced technologies has ushered in a new era in digital interaction, exemplified by the integration of the OpenCV library, NumPy, and the MediaPipe framework. OpenCV, a cornerstone in computer vision, empowers real-time video processing, while NumPy, a potent Python library, facilitates efficient numerical computing. Complementing these, the MediaPipe framework by Google provides a versatile solution for media processing tasks, from hand tracking to gesture recognition. This synergy forms the backbone of our exploration into the realm of computer vision, a field at the intersection of artificial intelligence and computer science. With a focus on extracting meaningful insights from digital images and videos, our project harnesses algorithms and techniques inspired by the human visual system.

Delving into the specifics, the integration of the OpenCV library unfolds a capability to capture live video feed from webcams, laying the foundation for real-time visual data processing. Transitioning seamlessly, the MediaPipe library takes the spotlight for hand tracking and landmark detection, identifying crucial hand landmarks, including fingertips and joints.The narrative extends into the nuanced world of gesture interpretation, where the program deciphers hand movements, mapping them to dynamic canvas interactions. Users can translate gestures into actions like drawing lines or selecting colors, fostering a seamless bridge between physical gestures and digital expression.

In the culmination of this technological orchestration, the canvas becomes a dynamic platform for digital art creation. Drawing upon computer vision techniques, the program enables users to paint and create in real-time, transforming hand gestures captured by the webcam into vibrant strokes on the virtual canvas. It exemplifies the potential of gesture-based interactions and digital art creation, showcasing the practical application of computer vision in our daily digital lives.

II. LITERATURE SURVEY

[1] Air Canvas Application Using OpenCV and Numpy Python:

This study introduces an innovative Air Canvas application leveraging OpenCV and NumPy. The primary focus is on fingertip detection for air writing. Deep Learning algorithms are employed to detect fingertips, generating a coordinate list for each frame. Two methods for creating a fingertip recognition dataset are explored: video-to-images and capturing photos with different backgrounds. The training involves the use of Single Shot Detector (SSD) and Faster RCNN pre-trained models, with Faster RCNN exhibiting superior accuracy. The study achieves a 94% accuracy for fingertip recognition.

[2] Virtual Air Canvas Application Using OpenCV and NumPy in Python:

This research delves into a Virtual Air Canvas application employing OpenCV and NumPy. Various models, including Neural Perceptron and Jordan's Recurrent Neural Network (JRNN), are examined for hand touch recognition and gesture detection. The proposed system incorporates Faster R-CNN for background subtraction and implements a centroid calculation algorithm for fingertip detection. Fingertip tracking utilizes the KCF tracing algorithm, converting detected fingertips into the HSV color space. The study showcases [4] the effectiveness of the proposed methods, achieving an impressive 99.2% acquisition rate and 99.0% accuracy in different hand positions.

III. EXISTING SYSTEM

The Virtual Air Canvas system, developed [8] using OpenCV and Python's NumPy library, enables users to draw freely in the air by moving their hands in front of a camera. It employs Neural Network models for color recognition and hand shape detection, achieving a high acquisition rate of 99.2%. The application focuses on fingertip detection, utilizing skin segmentation and background subtraction, followed by fingertip tracking with the aid of the Kernelized Correlation Filter (KCF) algorithm. The system's objective is [3] to create an intuitive and interactive platform for users to express their creativity through hand gestures in real-time.

IV. PROPOSED SYSTEM

The proposed system is a [10] virtual air canvas application developed using Python, OpenCV, and MediaPipe. It allows users to create drawings in the air using hand gestures captured by the webcam. The system employs hand landmark detection, fingertip tracking, and color selection functionalities, enabling users to draw on a virtual canvas in real-time. By recognizing specific hand gestures, users can switch between different colors and clear the canvas when needed.

V. ARCHITECTURE DIAGRAM

The entire process showed in the following architecture diagram. First the video stream acquisition done from webcam. Then using MediaPipe and OpenCV detecting process taken place to find hand and fingertips. Then draw the hand landmarks. The drawing shapes or characters depicts on the canvas.

VI. SYSTEM OVERVIEW

A. Webcam Capture and Processing

The Webcam Capture module [5] is responsible for accessing the video feed from the attached webcam. Utilizing the 'cv2' module from the OpenCV library, this module initializes the webcam, captures video frames in real-time, and displays them using the 'cv2.imshow()' function. The 'cv2.waitKey()' function ensures proper display window management, and 'cap.release()' terminates the webcam capture upon completion.

B. Hand Tracking

The Hand Tracking module is implemented using the MediaPipe framework, offering robust solutions for hand tracking and pose estimation. This module begins with library initialization, configuring parameters for hand detection. Landmark prediction is performed using a pre-trained machine learning model, enabling precise tracking of hand movements and positions. Gesture recognition analyzes predicted hand landmarks to interpret various hand gestures, ensuring real-time tracking for dynamic user interactions. The module visualizes detected hand landmarks on video frames, providing user-friendly feedback.

C. Canvas Interaction and Drawing

The Canvas Interaction and Drawing module [5] enable users to interact with a digital canvas through hand gestures. This module initializes the canvas with a white background, creating a graphical user interface for color selection, drawing tools, and canvas clearing. Color handling involves defining arrays for different colors, tracking index values for effective point management. Canvas drawing utilizes stored points for different colors, providing real-time visual feedback.

D. Display and Clean-up

The Display and Clean-up module manages the webcam output display and clean-up process after application closure. Display setup initializes the webcam, configures frame processing, and provides real-time display of video frames. The module includes functionality for terminating the application, releasing the webcam, and ensuring proper resource clean-up to maintain system stability.

Figure 6.1 shows the interface of the application, which contains a rectangular canvas where the user can draw [2] using hand gestures and text annotations for different color options and a "CLEAR" button on the canvas.

Figure 6.1 Application Interface

Figure 6.2 Hand Tracking

Figure 6.2 and Figure 6.3 shows the Real-time tracking of the user's hand movements and landmarks, which are visualized by points on the screen and the ability to draw lines on the canvas in different colors based on the user's hand movements and positions.

Figure 6.3 Canvas Interaction and Drawing

Figure 6.4 Display Cleanup

Figure 6.4 shows that the application will respond to the user's hand gestures, enabling them to change the drawing color and clear the canvas as needed. The output window will update in real-time as the user interacts with the application, providing a responsive drawing experience based on the detected hand gestures.

The actual output may vary [9] depending on the specific hand movements, gestures, and colors selected during the interaction. The application will continuously update the displayed [3] canvas and drawing in response to the user's hand movements, allowing for real-time drawing and interaction with the virtual canvas.

VII. FUTURE ENHANCEMENT

In future we can implement more drawing tools, such as shapes, brushes, and erasers, to make the application more versatile. And also, we can enable the option to save and load drawings to and from a file to provide a more comprehensive user experience. We can add the ability to change the thickness of the drawn lines based on finger movement or gestures. Finally, we can implement a more robust and efficient hand tracking algorithm to improve accuracy and responsiveness.

VIII. CONCLUSION

In conclusion, the integration of computer vision libraries, specifically OpenCV and MediaPipe, empowers the creation of dynamic, real-time interactive applications. This system adeptly captures and interprets hand gestures, [3] translating them into engaging drawing actions on the canvas. Its modular structure ensures easy maintenance and facilitates future enhancements, showcasing the potential for [9] a variety of applications in virtual art creation, education, and entertainment.

REFERENCES

[1] Asst. [2] Prof. Jahnavi S, K Sai Sumanth Reddy, Abhishek R AbhinandanHeggde, Lakshmi Prashanth Reddy,

(2022) "Virtual Air Canvas Application using OpenCV and NumPy in Python" in IJARIIE, Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka, INDIA, Vol-8 Issue-4, ISSN(O)- 2395- 4396.

[2] D. Vijendra Kumar, G. Vijaya Raj Siddarth, R. Venkata Satya Sravani, I. VishnuVardhan Reddy, Y. Lalitha Sri Naga DurgaVyshnavi, (2022) "Building an Air Canvas using NumPy and OpenCV in Python" in IJMTST, Godavari Institute of Engineering and Technology(A) Canvas using NumPy and JNTUK, Kakinada, ISSN: 2455- 3778, DOI: https://doi.org/10.46501/IJMTST08S0528.

[3] 6 Prof. S.U. Saoji, NishthaDua, Akash Kumar Choudhary, Bharat Phogat, (2021) "Air Application using OpenCV and NumPy in Python" in IRJET, (Deemed to be University) College of Engineering, Pune, Volume: 08 Issue: 08, e-ISSN: 2395-0056, p-ISSN: 2395-0072.

[4] 1 P. Ramasamy, G. Prabhu, and R. Srinivasan, (2016) " An economical air Writing system is converting finger movements to text using a web camera," 2016 system is converting finger movements to text using a web camera," 2016 Chennai, pp. 1-6.

[5] Sayli 4 More, Prachi Mhatre, Shruti Pakhare, Surekha Khot, (2022) "Air Canvas: Draw in Air" in IRJET, A.C Patil College of Engineering, Navi Mumbai, India, Volume: 09 Issue: 02, e-ISSN: 2395-0056, p-ISSN: 2395-0072.

[6] Sumedh Bansode, Saurabh Varkhad, Santosh Dhaigude, Shivam Waghmare, Prof. Shubhangi Suryawanshi,

(2022) 7 "Computer Vision Based Virtual Sketch Using Detection" in IJRASET, DR D Y Patil Institute of Technology, Pimpri, Pune, Volume 10 Issue V, ISSN: 2321-9653.

[7] 1 Y. Huang, X. Liu, X. Zhang, and L. Jin, (2016), " 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.

Sources

1	https://ijercse.com/viewabstract.php?id=16028&volume=volume9&issue=issue9 INTERNET 4%
2	https://www.ijraset.com/research-paper/virtual-drawing-board-drawing-and-writing-using-hand-gestures INTERNET 1%
3	https://www.momentslog.com/development/android/building-android-drawing-apps-implementing-canvas-and-drawing-functionality INTERNET 1%
4	https://www.irjet.net/volume9-issue02-feb-2022 INTERNET 1%
5	https://www.coursera.org/articles/ui-design INTERNET 1%
6	https://www.researchgate.net/profile/Saurabh-Saoji-2/publication/354142408_AIR_CANVAS_APPLICATION_USING_OPENCV_AND_NUMPY_IN_PYTHON/links/612721141f50fb262ff1a62c/AIR-CANVAS-APPLICATION-USING-OPENCV-AND-NUMPY-IN-PYTHON.pdf?origin=publication_detail INTERNET 1%
7	https://engg.dypvp.edu.in/computer-department-project.aspx INTERNET 1%
8	https://ijcrt.org/papers/IJCRT2305425.pdf INTERNET <1%
9	https://blog.research.google/2019/08/on-device-real-time-hand-tracking-with.html INTERNET <1%
10	https://www.researchgate.net/publication/370007226_Virtual_Air_Canvas_Using_OpenCV_and_Mediapipe INTERNET <1%

EXCLUDE CUSTOM MATCHES	OFF
EXCLUDE QUOTES	OFF
EXCLUDE BIBLIOGRAPHY	OFF