2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)
Bangalore, India. Oct 7-9, 2022

Virtual Canvas for Interactive Learning using OpenCV

Palak Rai
Computer Engineering
Don Bosco Institute of Technology
Mumbai, India
palakrai1101@gmail.com

Reeya Gupta
Computer Engineering
Don Bosco Institute of Technology
Mumbai, India
guptareeya53@gmail.com

Vinicia Dsouza
Computer Engineering
Don Bosco Institute of Technology
Mumbai, India
viniciadsouza00@gmail.com

Dipti Jadhav
Computer Engineering
Don Bosco Institute of Technology
Mumbai, India
dipti.dbit@dbclmumbai.org

***Abstract*— The purpose of this study is to discuss the usage of computer vision in educational applications. In recent years, air writing has become one of the most challenging and exciting research areas in image processing and pattern recognition. The project employs object tracking techniques to construct a motion-to-text converter that might be used as software in the field of education to allow students and teachers to write in the air. The project generates letters on the screen by utilizing computer vision to trace the path of a finger or an object. The resulting text can be utilized for a variety of things, including solving graph-related challenges. Questions on logical reasoning (Mathematics), to write formulas or execute derivations, Kindergarten kids are being taught the alphabet, to replace chalk and board (dustless classroom) in offline classes, etc. For the deaf, it will be a strong way of communication.**

Keywords—Air writing, gesture, object tracking, color detection, image processing, Computer Vision.

I. Introduction

The conventional art of writing is being supplanted by digital art in the digital age. Many different methods of writing are used in digital art, such as using a keyboard, a touch-screen surface, a digital pen, a stylus, or electronic hand gloves. However, in this system, we utilize object tracking using color detection techniques with the help of Python programming language and its various modules like OpenCV, NumPy, tkinter, keyboard to produce a natural man-machine interface. With the growth of technology, the demand for natural "human-computer interaction" solutions to replace traditional systems is fast growing. So, introducing artificial intelligence-based software in our education system which tracks the gesture of the user in real time and displays the output with negligible delay time will make an impact on the environment by reducing the use of plastic based markers replacing dust-filled classroom to dustless classroom and making traditional teaching style more fun and interactive.

In this work, we propose a simple, enjoyable, and helpful system that allows users to write on a screen just by waving their colored finger or a colored object in the air. While the idea of waving fingers in the air to draw on a screen without touching it may sound odd at first, the idea is quite feasible when computer vision and Python are combined with object tracking using color detection techniques.

After the pandemic caused in the year 2020 online mode of education (e-learning) has become one of the most important means of teaching and learning and hence one of the applications of Virtual Canvas is making online lectures more fascinating and fun by acting as a white board on which lecturers can write or draw in real time with negligible delay time and observe their gesture in the canvas (live window) with various user-friendly features making online lectures more fun and interactive. Another application of Virtual Canvas in education industry can be a need of dustless classroom during offline lecture, since chalk is made of calcium carbonate and calcium sulfate which has an impact on human health the user can face various health issue like allergies, itchy or watery eyes, respiratory problem and many more. Also, use of plastic based marker and boards, plastic made pen has an adverse effect on environment which in return increases global warming which can have a serious impact on not only human but on all the living organism. And thus, Virtual Canvas can contribute to the chain of reduce, reuse and recycle. Hence, there are numerous scenarios in which the system can be quite useful. For example, in the sphere of education, long PowerPoint presentations can be replaced with a virtual canvas to increase interactivity between students and to incorporate enjoyability within students. Using AI based Virtual Canvas can increase curiosity among students to develop and can encourage innovation among them. This method has the potential to be extremely useful in online as well as offline conversations.

II. Background

A. Literature Review

Around 10 literature articles were reviewed to obtain background knowledge on the vital concepts of detection and masking technology and to explore what has been done already in these respective areas. To track the real time moving object in videos OpenCV is used along with Convolution Neural Network and SVM classifier [2]. Using different deep learning methods for object detection face detection and pedestrian detection was discussed in [1]. The conclusion was made by modifying RCNN for object tracking which gave best results. Image processing technique, that is to determine the shape, color and contour of the object [4] for easy object tracking so that when two shapes overlap the contour of the biggest shape is selected. To enhance this color selection, segmenting the colors into different parts and then filtering them will help in better detection [8]. After processing, background subtraction is required to separate the foreground from background. For this Mixture of Gaussian method was discussed in [3] for detecting moving objects in outdoor environment. Gestures are the actions that the user makes that will be captured. A brief about gestures is discussed in [5]. To recognize the gestures, papers [6] and [7] gives brief review of methods that can be used. Making use of hardware device such as Raspberry Pi for object tracking and gesture recognition is discussed in [9]. This makes use of the raspberry pi keys for features to change brush color and size. The applications of Air Writing to solve real world problems using RCNN and YOLO is proposed in [10].

B. Survey

A survey was conducted to know what all challenges were faced by teachers and students during online teaching and learning period. About 8 out of 10 teachers said they faced issues where they could not teach students in traditional method that is using a white or black board. Because of this teacher had to adapt to teaching using devices such as stylus or notepad where they could solve mathematical equations or formulae. But the main problem here was that these tools were not provided by the schools or colleges as some were not financially funded well. So, teaching using shared images, PDFs or presentations came in. This created lack of interactivity between students and the motivation of learning within students decreased as there was no fun way incorporated. Also, if the student had to explain the idea or solve a numerical, they would need to solve it in their books and then scan their page and send it to their teacher or upload it on school websites. This resulted in wastage of time and increased copying tendencies as the teacher didn't know who actually solved the numerical. These were some of the problems faced by school and college teachers.

When we consider preprimary school teachers, they had issues related in teaching alphabets or numbers to students, as gestures play a very important role here. Following the gestures is how a preprimary school teacher chooses to teach an alphabet.

Another survey was conducted for the students about their overall learning experience in online mode and there was a 50-50 positive-negative response. But the common issue was the increased number of assignments because teachers could not make the students solve them during lecture slots.

C. Aim and Objective

The aim of our project is to make an inexpensive product that lets the user write in air using any object with ease and to improve the online learning experience by solving problems that need the use of expensive products such as stylus or touch screen.

The objective following this aim is to make Virtual Canvas user friendly by incorporating features to save, undo, maximize/minimize brush size and whiteboard for cluttered backgrounds. Another objective is to create an environmentally friendly product which decreases the use of chalk or colored markers which are considered harmful for the environment as well as human health.

D. Requirements

- The model is created using the Python programming language.
- The gestures will be controlled using OpenCV (Open Source Computer Vision) which makes use of a video camera (laptop or computer) to track the user gestures.
- NumPy is used for signal processing and also for some numerical operations.
- Tkinter is used to create the user interface at the start of the software and to select the object color.
- The keyboard will be used to track the user inputs or shortcuts to save, undo, change color and clear the screen.

III. PROPOSED SYSTEM

The project uses the Open source Computer Vision library to capture gestures as well as to do Image Segmentation and Image Processing [5]. The models flowchart from start to end can be seen in Fig.1.

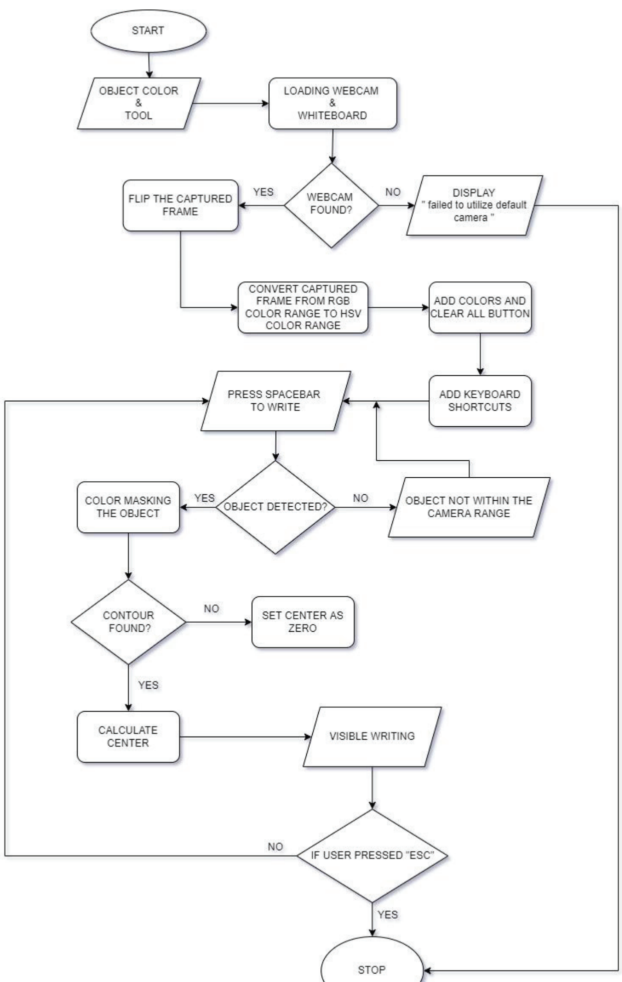

Fig. 1. System Flowchart

Describing the flowchart, the first step for the user is to select the color of the object. The range of the color selected is passed, the live video is captured and then the captured frame is flipped since it is a mirror image. The captured video is then converted into HSV color range. Once the live frame appears on the screen all the color buttons and clear button is displayed on the live frame along with the keyboard shortcuts. For the user to write in the air through their gesture, the model will detect the object using the spacebar key. If the object is not detected a display message appears saying “object not within the camera range" so the user will have to bring the object within the camera range and will have to press the space bar key again. Once the object is detected color masking is done followed by the background subtraction. Next step is to find the contour. Contour is an object with same color intensity and then the centre is found on the object. If contour is not detected the centre is set to zero else the centre is found on the contour. Followed by this the user can see the output on both the screen that is the live frame and the white board. If the user wants to exit Virtual canvas then they will need to press esc key and the system closes or stops running.

IV. System Methodology

This section gives details about the different parts of the model. Starting from IV-1 which is the Graphical User Interface needed for selecting the object color with the help of arrays followed by Image Segmentation that includes frame reading and background subtraction [3], Image Processing which is detecting the contour [4] for object detection and finally the output which is draw on canvas that makes use of keyboard keys. Fig.2. is the architecture of the model

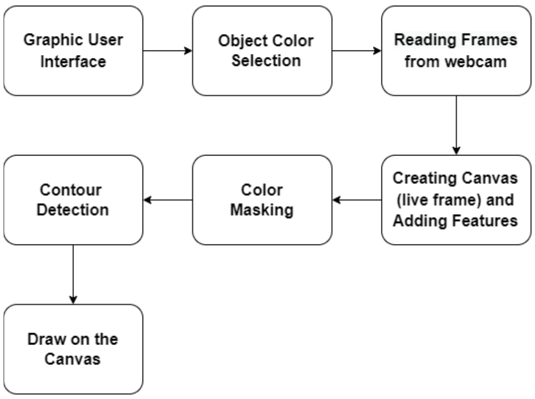

Fig. 2. System Architecture

A. Graphic User Interface

An interface is made using tkinter library in python. This interface has all details regarding what virtual canvas is and its manual for easy usage of the software along with option to select the object color.

B. Object Color Selection

Once the user selects the Color of the object of his/her choice then that color is passed to the model for color masking and contour detection. This ensures that the model detects and tracks the user specified color only.

The color of the object is detected using the Hue saturation value which is difficult and is affected by various parameters one of which is sunlight. Also, it is not easy to have one particular colored object available with the user so restricting the user to have just one colored object is not a good idea. So, the model allows user to select four different colors (Blue, Green, Red and Yellow) for the object color selection.

Initially when the Frame is captured it is captured in Blue (255, 0, 0), Green (0, 255, 0), Red (0, 0, 255) that is BGR color space to detect the colored object easily and quickly into Hue saturation value color space [2]. Various function like *cv2.cvtCOLOR(), cv2.inRange(), Cv2.dilate()* were used to converts the color-space, and create binary segmentation and to increase segmentation respectively [8]. After this the upper Hue saturation value and lower Hue saturation value of four different colors (Blue, Green, Red and Yellow) are stored in NumPy arrays (np.array).

C. Frame Reading

To capture the video in real time *cv2.VideoCapture* is used. This is a module of the OpenCV library. It creates a video object that would help stream or display the video. The video captured is mirror video which is then flipped by the model using cv2.flip. Also, the captured frame is in BGR color space which is then converted into HSV color frame [8].

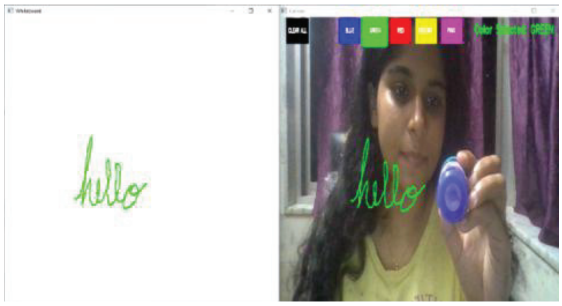

Fig.3. Whiteboard and Canvas windows

Sometimes background of live frame isn’t suitable for displaying the written text (output) so access to a whiteboard was necessary. The whiteboard displays the written text (output) clearly on a white background. It is in sync with the canvas (live window) so the delay time is 0.

This is done with the use of np.zeros which creates a white image that acts as the background to the whiteboard. The parameters passed to this function are the dimension of the white image (whiteboard window). Fig.3. shows the whiteboard along with the canvas window. The option of resizing the frame as per requirement is done using the *CV2.Window_Normal*

D. Canvas Creation

A Canvas (live frame) is created along with a white board which is in sync with the canvas using the function np.zeros [10]. Various feature like color buttons, clear button is displayed on the canvas(live frame) as seen in Fig.4. using the function *cv2.rectangle, cv2.putTextalong* with various functionality.

Fig.4. Display buttons on the screen

Five different color buttons are provided at the top center of the Canvas Window (live frame) which helped user to easily switch between the five colors (Blue, Green, Red, Yellow, and Pink) according to the users needs. Selected color appears slightly bigger than the unselect color.

To store and handle the color points of the five colors (Blue, Green, Red, Yellow, Pink) five different arrays were created along with color index (initially set to 0) which is used to mark points of a specific array of that particular color [10]. The Maximum length of the color point of the colors is set as 1024.

At the top left-corner of the Canvas Window (live frame) clear button is placed for easy access. Clear button clears all the writing in both the screen (canvas and whiteboard) before or after saving the writing (depends on the user).

The five different arrays that were created to store and handle the color points of the five colors (Blue, Green, Red, Yellow, Pink) are all dequeued and the color index of the five colors (Blue, Green, Red, Yellow, Pink) are initialized to 0

E. Color Masking

Color masking process creates mask of the object using the function *cv2.erode* which creates a boundary of the object detected and fills it with white [3] and then background subtraction is performed simultaneously where the background is changed to black using the function *bgModel.apply* [8].

F. Contour Detection

The model then detects the contours which is the shape of the object found with similar HSV color intensity. This is done using *cv2.findContours*. As there is a possibility of finding many contours the model will find the biggest contour among all the contours found [4].

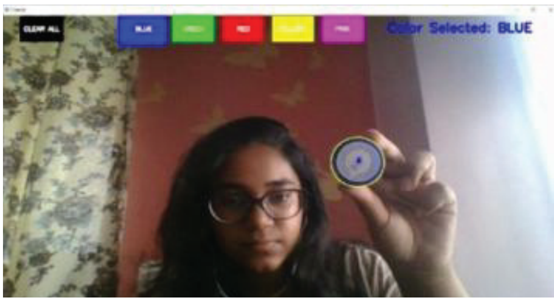

Fig.5. Contour Detection

This helps in reducing the errors regarding same background and object color. Then with the help of cv2.circle a circle is formed over the object and then the center (C) of the object is calculated as seen in Fig.5. using the formula (1)

$$C = \frac{m10}{m00}, \frac{m01}{m00} \qquad (1)$$

Here, C = mass center

m10, m01, m00 = spatial moments

G. Draw on Canvas

After completing the above steps, the user can finally draw and write in the air easily and the model will track the object and will form points at each position for the air canvas effect. Then the model will display the video frame by frame. The function used here is imshow().

H. Keyboard Keys

Filling the canvas (live window) with least used buttons will occupy a lot of space which leaves with less writing space. So, for better user interactivity shortcuts keys were provided. In total there are 6 shortcut keys, they are – (z) to undo changes, (+) to increase brush thickness,(-) to decrease brush thickness, (numbers 1 to 5) to change brush color, (s/clt s) to save and (q/esc) to quit the program. These features work with the help of the keyboard module in python. Whiteboard is in sync with the canvas (live window) so the delay time is 0.

To start writing in the air and for the model to detect the object and draw points at each and every position on the canvas and the whiteboard, user needs to press the spacebar key. The gestures will be detected and then the output would be displayed.

Fig.6. Three saving modes

Fig.6. shows the save shortcut key and its 3 modes – Canvas, whiteboard, blackboard. The user can save in any of the 3 modes. The images saved will be in PNG format at the user's desired location.

V. EXPERIMENTS

A. Experiment Details

The model was tested in different backgrounds with different light settings. The backgrounds included indoor and outdoor environments. In indoor environments the model was tested with a plain white background and with cluttered background that included multiple colors in the background. Three types of lighting was considered that is very bright (brightness intensity 80%), normal (brightness intensity 40%) and very dark (brightness intensity 5%). The accuracy rate is generated by testing the model for 5 minutes by using all the features and different color objects.

For outdoor environments the backgrounds were a train platform where people were moving constantly and a still background that is in a garden with less movement in the background. The lightings considered are very bright and very dark. This is because in outdoor environments lighting cannot be adjusted. Two cameras with different Megapixels are used to indentify best accuracy.

B. Results and Discussion

The movement of the object is detected and is visible on the video screen with five different colors buttons, a clear all button and display message of the selected color at the top of the screen. Also, the whiteboard is in sync with the canvas. There are a number of options to pick from which include various system modes that serve various functionalities. Basic Functionality of the model include

- Virtual Canvas -The system will trace the movement of the object and display them in real time.
- Whiteboard - The system will trace the movement of the object and display them in whiteboard

- Color options - The user can select a different color for the text from the available options.
- Clear all – Clear all clears the characters on the screen and provides a clean space to write on.

The model gives different accuracy rates when placed in different backgrounds. Tables 1 and 2 give a view of the accuracy rates in different environments.

TABLE I. ACCURACY RATES FOR INDOOR ENVIRONMENTS

Test No.	Camera Quality	Back-ground	Lighting	Accuracy (%)
1	1.3 MP	Plain White	Very Bright	94%
2	1.3 MP	Plain White	Normal	98%
3	1.3 MP	Plain White	Very Dark	91%
4	1.3 MP	Cluttered (many objects)	Very Bright	90%
5	1.3 MP	Cluttered (many objects)	Normal	98%
6	1.3 MP	Cluttered (many objects)	Very Dark	86%
7	5 MP	Cluttered (many objects)	Very Bright	94%
8	5 MP	Cluttered (many objects)	Very Dark	92%

TABLE II. ACCURACY RATES FOR OUTDOOR ENVIRONMENTS

Test No.	Camera Quality	Back-ground	Lighting	Accuracy (%)
1	1.3 MP	Crowded (People walking)	Very Bright	81%
2	1.3 MP	Crowded (People walking)	Very dark	84%
3	5 MP	Crowded (People walking)	Very Bright	89%
4	5 MP	Crowded (People walking)	Very dark	91%
5	1.3 MP	Plain	Very Bright	94%
6	1.3 MP	Plain	Very Dark	91%
7	5 MP	Plain	Very Bright	94%
8	5 MP	Plain	Very dark	92%

From the experiments conducted we can conclude that the model gives accurate results in plain background whether that is in indoor or outdoor environments. The quality of the camera plays a very important role for accuracy when background is cluttered. For outdoor environments with crowded background the accuracy can be increased if the camera quality is increased.

VI. CONCLUSION

The user is able to draw and write on the screen with ease on a virtual canvas, allowing them to effectively interact with others. This technology has the ability to put standard writing methods to the test. We covered various types of technologies that could be employed to construct this system. Virtual canvas works well with a good quality camera as well as a clear background. Few challenges were faced if the room is exposed to too much sunlight or in cluttered background which results in weak object color detection . Excluding these challenges Virtual canvas strikes to be an excellent software for interacting with the digital environment. This model can now produce the output that our group had envisioned, and it will be enhanced in the future to include new features and improve efficiency.

REFERENCES

[1] Rajesh D, Yamini C “A Review on Object Detection using OpenCV” in International Journal of Innovative Technology and Exploring Engineering (IJITEE)

[2] Suraj S, Shende, Vikas S, Bharti P “Real Time Object Detection using Deep-Learning and OpenCV” in International Research Journal of Engineering and Technology (IRJET)

[3] Mohamed, Shahrizat Shaik, Nooritawati Md Tahir, and Ramli Adnan. “Background modelling and background subtraction performance for object detection.” 2010 6th International Colloquium on Signal Processing & its Applications. IEEE, 2010.

[4] Raghav P, ArchitG , Manas S, in “Contour, Shape & Color Detection using OpenCV-Python” in International Journal of Advances in Electronics and Computer Science

[5] Mitra, Sushmita. Senior Member, IEEE, and Tinku Acharya, Senior Member, IEEE,“Gesture Recognition: A Survey”. IEEE Transactions On Systems, Man, And Cybernetics—part C: Applications And Reviews 37.3 (2007): 311.

[6] Chen, Mingyu, Ghassan AlRegib, and Biing-Hwang Juang. “Air-writing recognition—Part I: Modeling and recognition of characters, words, and connecting motions.” IEEE Transactions on Human-Machine Systems 46.3 (2015): 403-413.

[7] Chen, Mingyu, Ghassan AlRegib, and Biing-Hwang Juang. “Air-writing recognition—Part II: Detection and recognition of writing activity in continuous stream of motion data.” IEEE Transactions on Human-Machine Systems 46.3 (2015): 436-444.

[8] Vandit G “Real Time Color Segmentations and Color Filtration” in IJSRD National Conference on Advances in Computer Science Engineering & Technology May 2017

[9] D. U. Lakshmi and B. Harish, "A novel air writing recognition system using Raspberry Pi for the control and interaction of digital systems," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 2017

[10] S.U. Saoji, Nishtha D, Akash C, B Phogat “Air canvas application using Open CV and numpy in python” in (IRJET) Volume: 08 Issue: 08 | Aug 2021