

# Software Group Project

Team Members:

**1) RUTVIK BALAR (19IT006)**

Air canvas

**Virtual painter to draw in air.**

**Abstract:**

Air canvas is a paint app that draws on screen by detecting and tracking specific color object. In this pandemic it is very difficult to study online and it's also boring work so using this virtual painter we can increase interested rate in student. this is a virtue paint and using this we can draw and erase in air just detecting particular color or object . Given the real time webcam data, this paint-like python application uses OpenCV library to track an object-of-interest and allows the user to draw by moving the object, which makes it both awesome and challenging to draw simple things.

# Introduction:

Ever wanted to draw our imagination by just waving our finger in the air. So, we have built an Air Canvas which can draw anything on it by just capturing the motion of a colored marker with a camera. Here a colored object at the tip of the finger is used as the marker.

We will be using the computer vision techniques of OpenCV to build this project. The preferred language is Python due to its exhaustive libraries and easy to use syntax but understanding the basics it can be implemented in any OpenCV supported language.

Here Color Detection and tracking are used in order to achieve the objective. The color marker is detected and a mask is produced. It includes the further steps of morphological operations on the mask produced which are Erosion and Dilation. Erosion reduces the impurities present in the mask and dilation further restores the eroded main mask.

Steps -

Step 1: Find the color range of the target object and save it.

Step 2: Apply the correct morphological operations to reduce noise in the video

Step 3: Detect and track the colored object with contour detection.



Step 4: Find the object’s x,y location coordinates to draw on the screen.

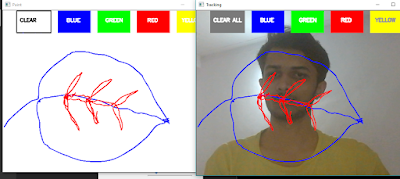
Step 5: Add a Wiper functionality to wipe off the whole screen.

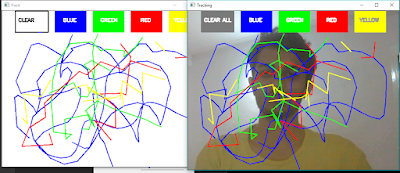
Step 6: Add an Eraser Functionality to erase parts of the drawing.

I’ve designed the pipeline of this application in such a way that it’s easily reusable for other projects, for e.g. if you were to make any project that involved tracking a colored object then you can use steps 1-3 for that. Also this breakdown makes it a lot easier to debug a step once you’re running this for yourself as you’ll know exactly which step you got wrong. Each single step can be run independently.

Output –

Here we have try to draw a leaf. My marker is a blue-colored dart.

[](https://1.bp.blogspot.com/-Lbjts6QImlw/YFK-jwTXv9I/AAAAAAAADMc/-xmdxQGJyykv1BRmZ-j1hTJMOtjsnVDkACLcBGAsYHQ/s1163/Capture.PNG)

[[](https://1.bp.blogspot.com/-rqwLE_64RPM/YFK-waZqx1I/AAAAAAAADMk/v3yfUvk0rv0XX524yAinxfOQNpUtTC0cgCLcBGAsYHQ/s1212/Capture2.PNG)](https://1.bp.blogspot.com/-rqwLE_64RPM/YFK-waZqx1I/AAAAAAAADMk/v3yfUvk0rv0XX524yAinxfOQNpUtTC0cgCLcBGAsYHQ/s1212/Capture2.PNG)



# Literature Survey:

We will be using the computer vision techniques of OpenCV to build this project. The preferred language is python due to its exhaustive libraries and easy-to-use syntax but understanding the basics it can be implemented in any OpenCV supported language.

Here Colour Detection and tracking are used to achieve the objective. The color marker is detected and a mask is produced. It includes the further steps of morphological operations on the mask produced which are Erosion and Dilation. Erosion reduces the impurities present in the mask and dilation further restores the eroded main mask.

These were the large scale issues that most school and college face and to solve the problem, “Air canvas” was developed to help with latest technology.

# Technical Specifications:

### Target Platform:

* Web Application.

### Tech Stacks to be Used:

* **Frontend:** Python3 ,Numpy , OpenCV.
* **Editor:** VS Code .

### Domains Covered:

* Image processing.
* Python
* OpenCV
* Numpy

### Github link:

* <https://github.com/rutvik-balar/>



# References:

* [OpenCV tutorials](https://towardsdatascience.com/tutorial-webcam-paint-opencv-dbe356ab5d6c)
* [All about python by code with harry](https://codewithharry.com/)
* [Air canvas by GFG](https://www.geeksforgeeks.org/create-air-canvas-using-python-opencv/)
* [OpenCV documentation](https://docs.opencv.org/master/d6/d00/tutorial_py_root.html)
* [Python](https://docs.python.org/3/)

# Conclusion:

This is a simple demonstration of the image processing capabilities of OpenCV. If you find this either interesting or easy or boring or think that there is nothing challenging about it, I urge you to try adding more paint features like an eraser, different types of brushes, color-fill options etc. to really get a feel of how fun OpenCV can be. We’ve also got experience with the Version Control System as we hosted our codebase on GIT. It was also beneficial to us as it helped us to show insights of project management while working with a team.