

## PRACTICAL - 5

**Name:** Shivanshu Anant Suryakar

**PRN:** 1841048

**Batch:** B3

**Class:** L.Y Computer Engineering

**Aim:** Python program to Color Detection using Pandas & OpenCV

---

### Theory:

Color detection is the process of detecting the name of any color. Simple isn't it? Well, for humans this is an extremely easy task but for computers, it is not straightforward. Human eyes and brains work together to translate light into color. Light receptors that are present in our eyes transmit the signal to the brain. Our brain then recognizes the color.

Colors are made up of 3 primary colors; red, green, and blue. In computers, we define each color value within a range of 0 to 255. So in how many ways we can define a color? The answer is  $256 \times 256 \times 256 = 16,581,375$ . There are approximately 16.5 million different ways to represent a color. In our dataset, we need to map each color's values with their corresponding names. But don't worry, we don't need to map all the values. We will be using a dataset that contains RGB values with their corresponding names.

### Some Real-world Applications

- In self-driving car, to detect the traffic signals.
- Multiple color detection is used in some industrial robots, to performing pick-and-place task in separating different colored objects.

This is an implementation of detecting multiple colors (here, only **red**, **green** and **blue** colors have been considered) in real-time using Python programming language.

**Step 1:** Input: Capture input using image file.

**Step 2:** Read the video stream in image frames.

**Step 3:** Convert the imageFrame in BGR( RGB color space represented as three matrices of red, green and blue with integer values from 0 to 255) to HSV(hue-saturation-value) color space. *Hue* describes a color in terms of *saturation*, represents the amount of gray color in that color and *value* describes the brightness or intensity of the color. This can be represented as three matrices in the range of 0-179, 0-255 and 0-255 respectively.

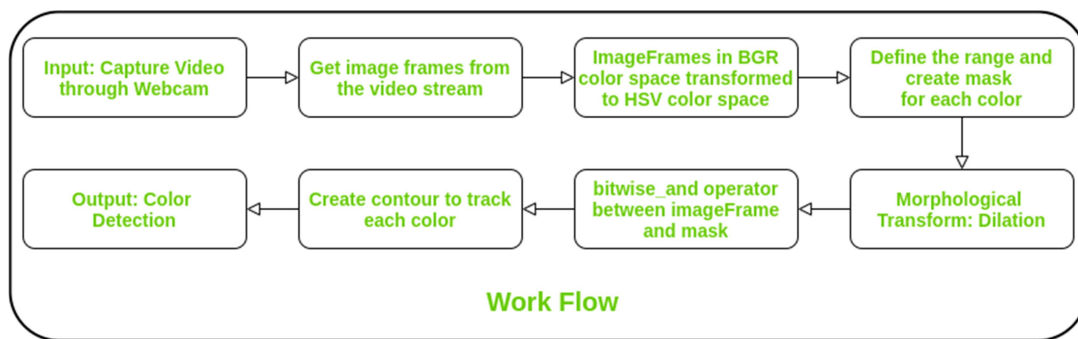
**Step 4:** Define the range of each color and create the corresponding mask.

**Step 5:** Morphological Transform: Dilation, to remove noises from the images.

**Step 6:** bitwise\_and between the image frame and mask is performed to specifically detect that particular color and discard others.

**Step 7:** Create contour for the individual colors to display the detected colored region distinguishly.

**Step 8:** Output: Detection of the colors in real-time.



### Requirements:

- Numpy
- Open CV

- Pandas

**Output:**



**Conclusion:**

In this practical we have studied Colour Detection using Pandas & OpenCV