```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

#Upper code are mandatory to be written if you want to run it in Python 2.x version environments

#Composer : Dongbin Kim
#Date : 2019-06-02
#Title : OPENCV Tutorial - Color Tracking

import cv2 #OpenCV Librarty


def callback(value):
    pass

def main():
    cam = cv2.VideoCapture(0)  # Create Camera Frame

    if cam.isOpened() == False:  # Camera Create Confirmation
        print
        'Can\'t open the CAM(%d)' % (0)
        exit()

    # Window creation, and size adjustment
    cam.set(3,320) #length
    cam.set(4,240) #width
    #cam.set(5,30) #framerate adjustment

    cv2.namedWindow('Trackbar') #Create a Trackbar to control the HSV values

    cv2.createTrackbar('Low H', 'Trackbar',22,179, callback) #Low Hue value
    cv2.createTrackbar('High H', 'Trackbar',37,179, callback) #High Hue value
    cv2.createTrackbar('Low S', 'Trackbar',123,255, callback) #Low Saturation value
    cv2.createTrackbar('High S', 'Trackbar', 255, 255, callback) #High Saturation value
    cv2.createTrackbar('Low V', 'Trackbar', 83, 255, callback) #Low Value value
    cv2.createTrackbar('High V', 'Trackbar', 255, 255, callback) #High Value value

    while (True):

        ret, src = cam.read() #Achieve image from the camera
        if not ret:
            break

        src_hsv = cv2.cvtColor(src, cv2.COLOR_BGR2HSV) #Convert source image(src) into HSV from BGR
        v1_min = cv2.getTrackbarPos('Low H','Trackbar')
        v1_max = cv2.getTrackbarPos('High H','Trackbar')
```

```
        v2_min = cv2.getTrackbarPos('Low S', 'Trackbar')
        v2_max = cv2.getTrackbarPos('High S', 'Trackbar')
        v3_min = cv2.getTrackbarPos('Low V', 'Trackbar')
        v3_max = cv2.getTrackbarPos('High V', 'Trackbar')
        finalimage = cv2.inRange(src_hsv,(v1_min, v2_min, v3_min),(v1_max, v2_max,
          v3_max)) #The value that adjusted on the trackbar will be applied

        cv2.imshow("cam", src) # Display the raw image on the Window "cam"
        cv2.imshow("final image", finalimage) #Display the HSV adjusted/filtered
          image on the Window "final image"


        if cv2.waitKey(1) >= 0:
            break

    cam.release()
    cv2.destroyWindow('CAM_Window')

if __name__ == '__main__':
    main()
```