МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**Дисциплина: Алгоритмы цифровой обработки мультимедиа**

Работу выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. В. Попов

Направление подготовки: 02.03.02 Фундаментальная информатика и информационные технологии

Преподаватель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Крамаренко

Краснодар

2024

**Задание 1**

Прочитать изображение с камеры и перевести его в формат HSV.

Для выполнения данного задания бала написана функция HSV\_recording(). Затем был написан метод VideoCapture(). Далее необходимо перевести изображение с камеры в формат HSV, это можно сделать с помощью функции cvtColor, указав флаг COLOR\_BGR2HSV (переводит из формата BGR в формат HSV).



На рисунке 1 показан результат первого задания.

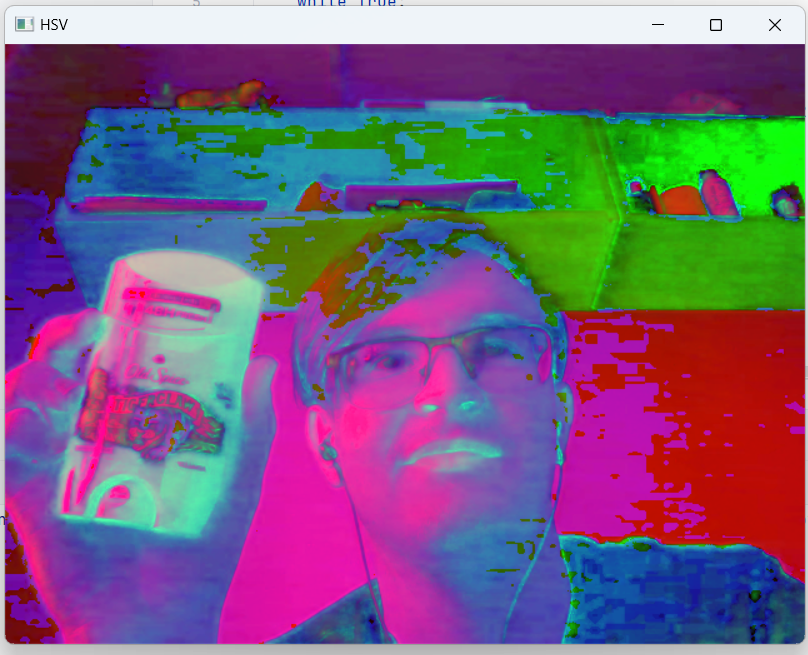
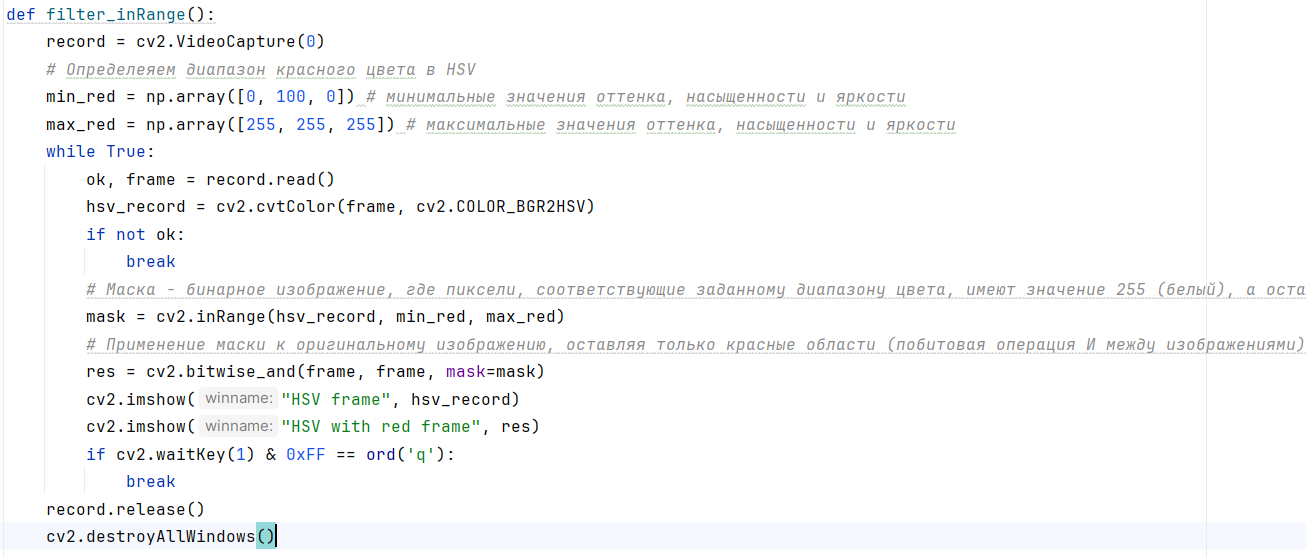


Рисунок 1 – Вывод результата в формате HSV

**Задание 2**

Применить фильтрацию изображения с помощью команды inRange и оставить только красную часть, вывести получившееся изображение на экран(treshold), выбрать красный объект и потестировать параметры фильтрации, подобрав их нужного уровня.

Была написана функция filter\_inRange(). Необходимо определить диапазон красного цвета, то есть какие цвета мы будем считать красным. Введём две переменные min\_red – минимальные значения HSV, которые мы будем считать красным цветом и max\_red – максимальное значение HSV. Функция inRange() на выходе дает набор пикселей со значением либо 0, либо 1, т.е. цвет удовлетворяет заданным границам или не удовлетворяет. Далее применяем маску к текущему кадру, с помощью лог. операции "И" для каждого пикселя.



Результат представлен на рисунке 2.

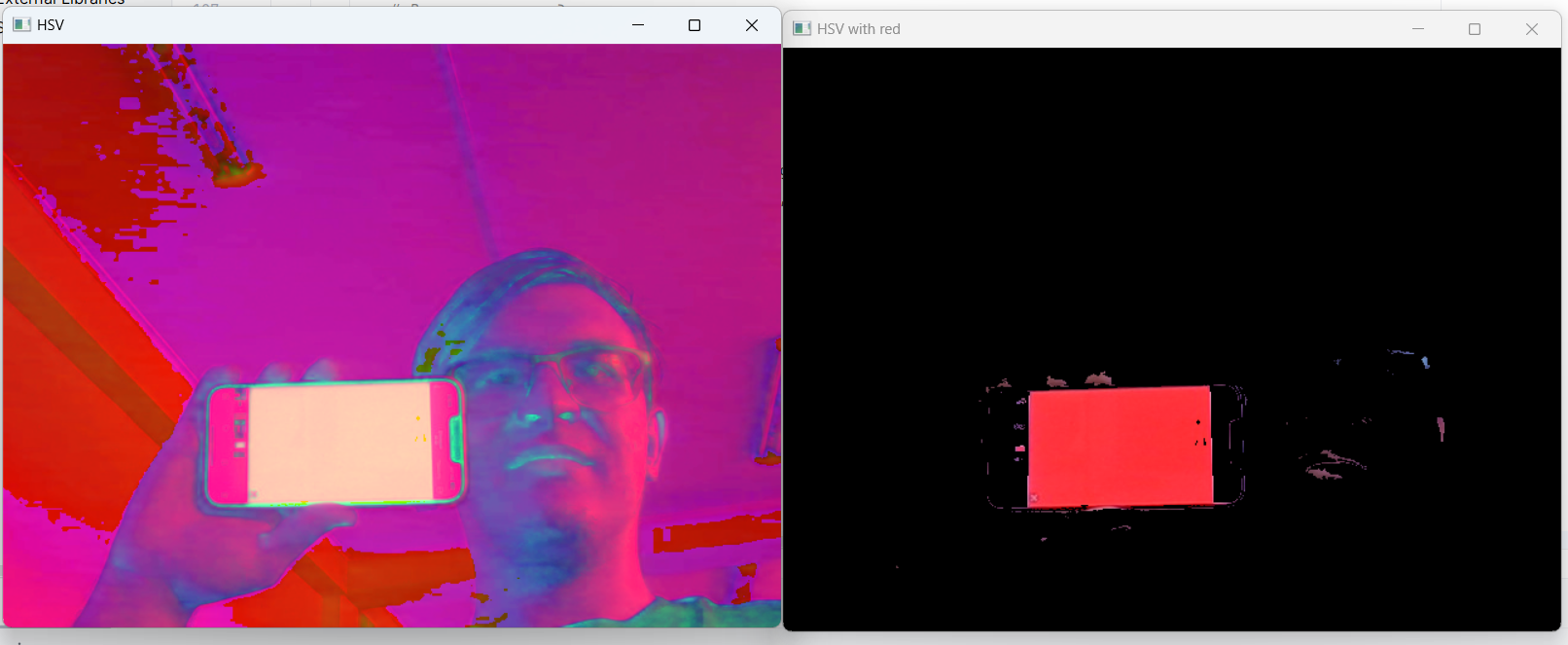


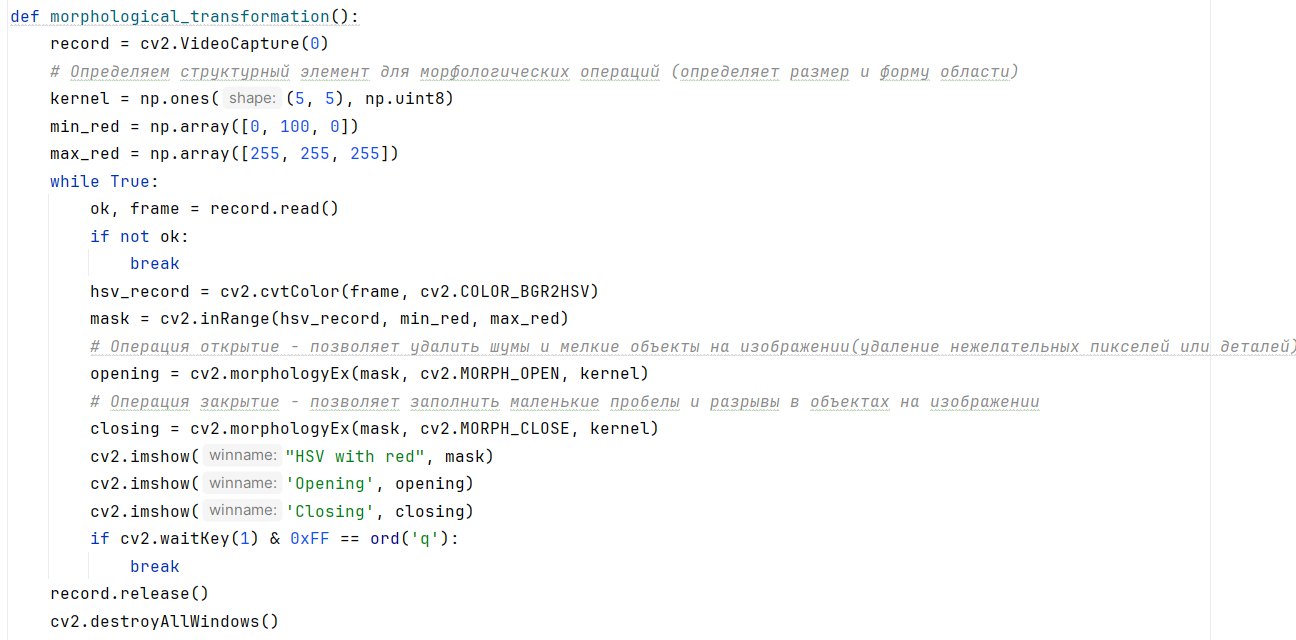
Рисунок 2 – Вывод изображений с флагами

**Задание 3**

Провести морфологические преобразования (открытие и закрытие) фильтрованного изображения, вывести результаты на экран, посмотреть смысл подобного применения операций erode и dilate.

Была написана функция morphological\_transformation(). Определяем структурный элемент для морфологических операций. При морфологическом преобразовании открытия изображения позволяет удалить шумы и других нежелательных пикселей. Морфологическая операция закрытия позволяет заполнить пиксельные разрывы между объектами на изображении. Таким образом, операция dilate позволяет расширять яркие области изображения. Операция erode наоборот, уменьшает область объекта на изображении. Морфологическое открытие - это последовательное применение операций erode, а затем dilate. Оно используется для удаления мелких шумов из изображения. Морфологическое закрытие - это последовательное применение операций dilate, а затем erode. Оно используется для заполнения небольших пробелов и разрывов

Необходимость этих операций обусловлена тем, что они помогают улучшить качество бинарных изображений, удаляя шум и улучшая структуру объектов



Результат представлен на рисунке 3.

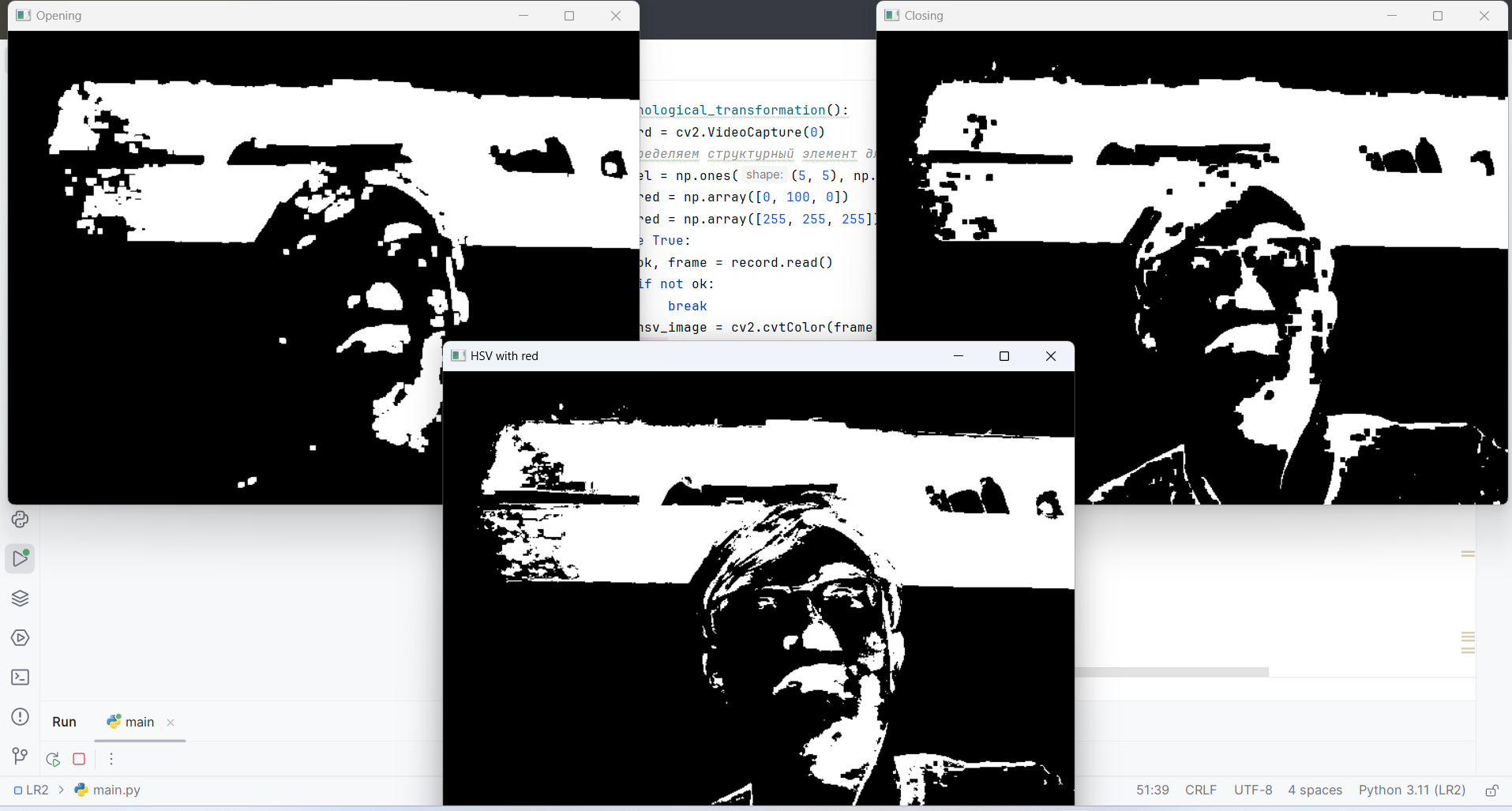
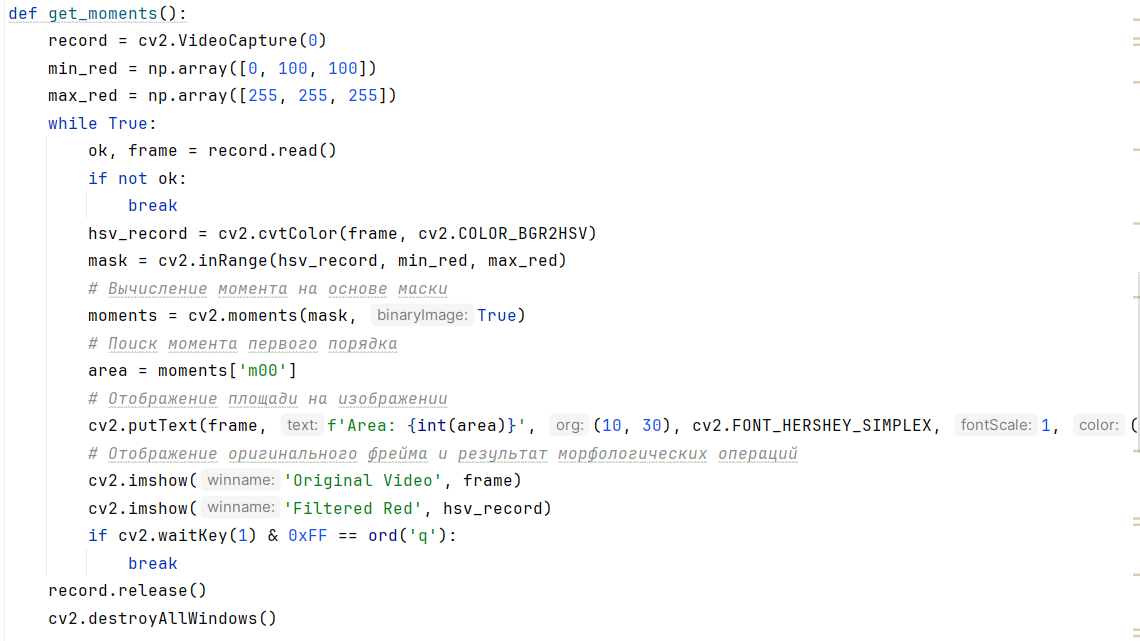


Рисунок 3 – Морфологическое преобразование

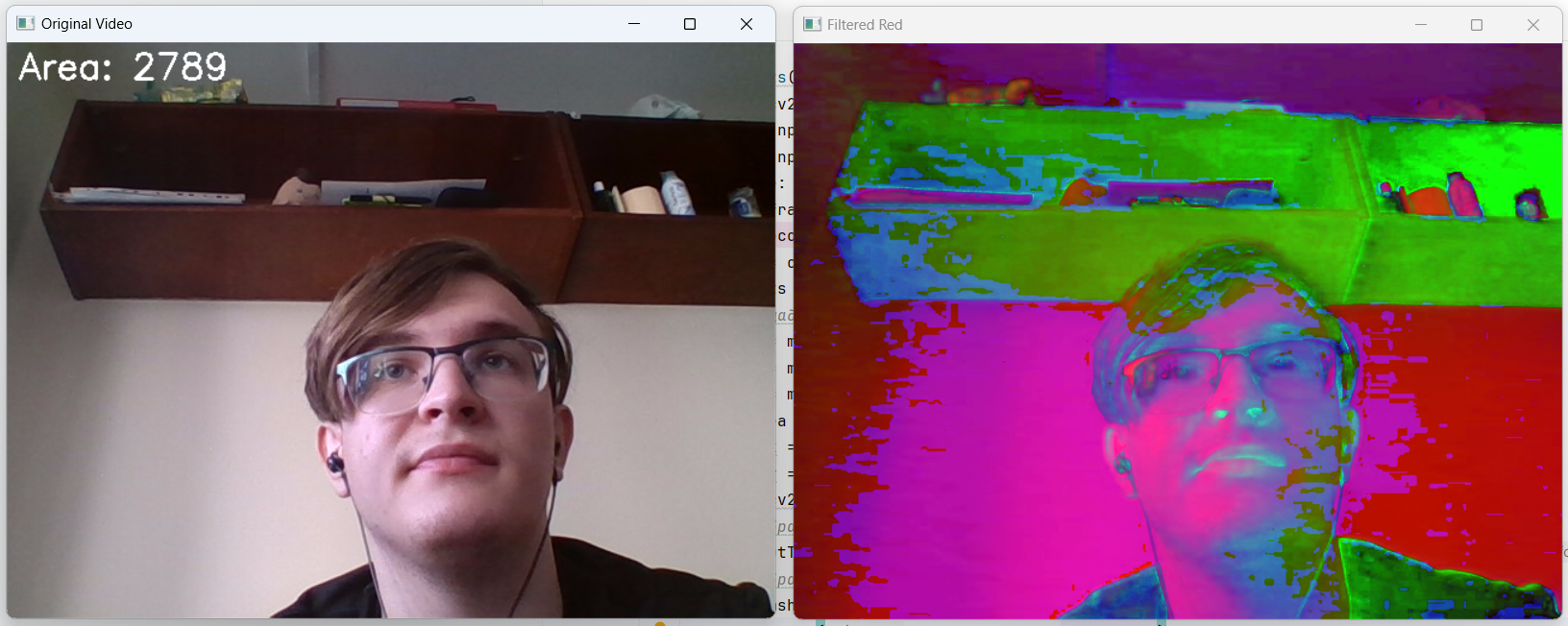
**Задание** 4

Найти моменты на полученном изображении 1 первого порядка, найти площадь объекта.

Была написана функция get\_moments(). Моменты изображения – определённые средневзвешенные значения интенсивности пикселей изображения. Момент нулевого порядка m00 — это количество всех точек, составляющих пятно. Воспользуемся стандартной функцией для вычисления моментов кадра cv2.moments(). Используем нулевой порядок m00. Выводим оригинальное и изменённое изображение на экран.



Результат представлен на рисунке 4



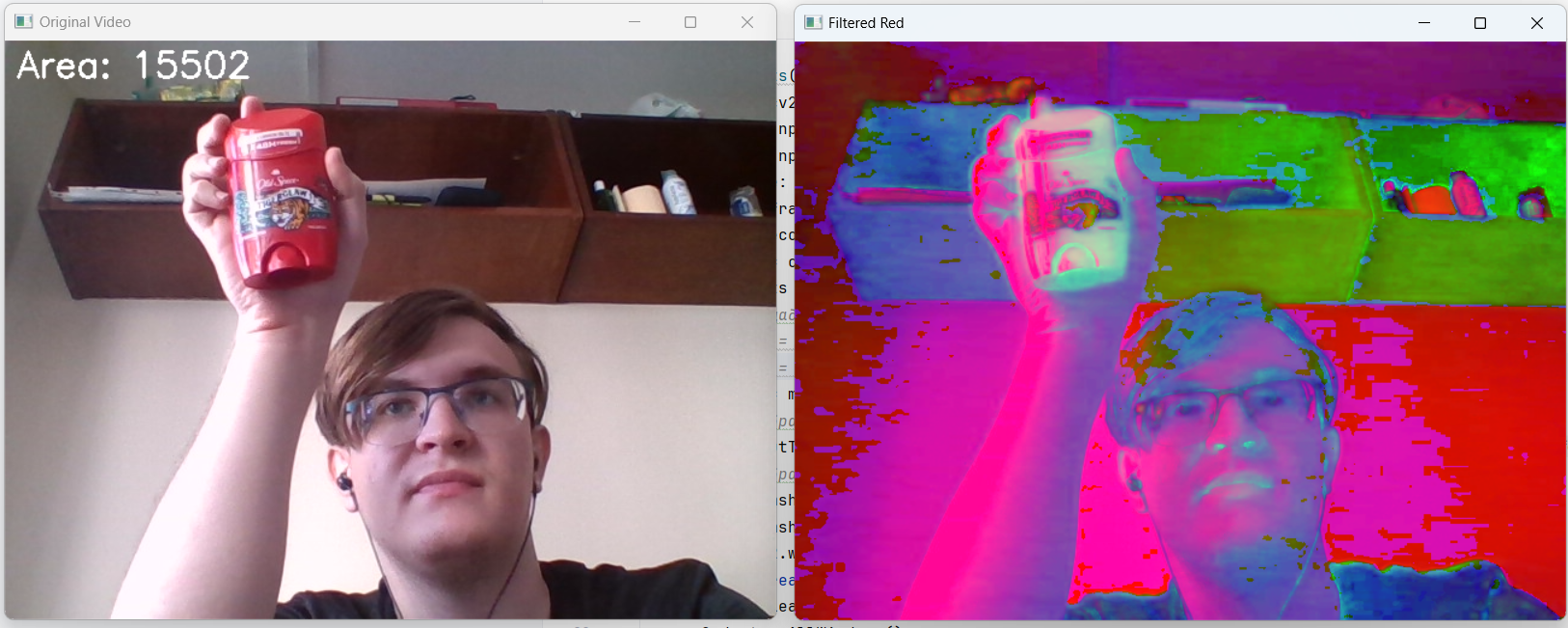
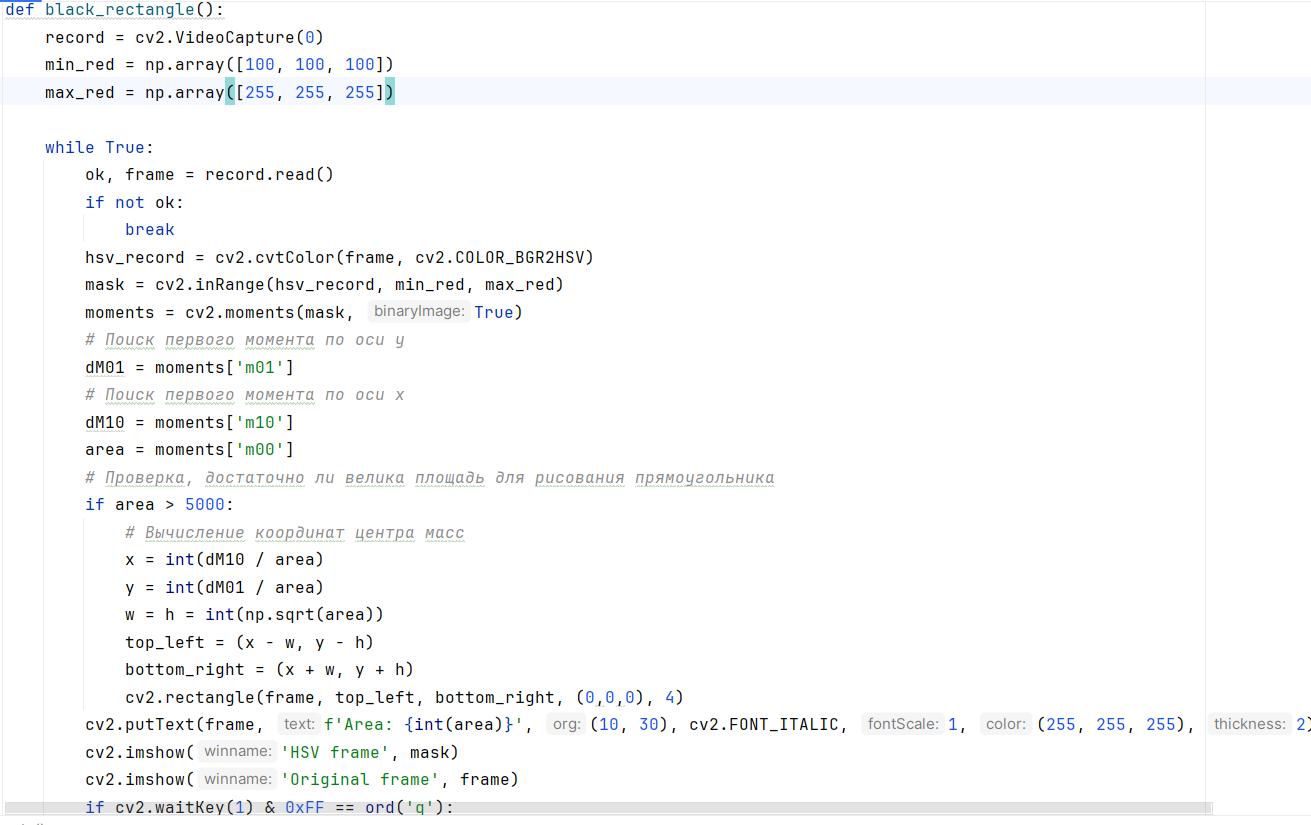


Рисунок 4 – Момент изображения

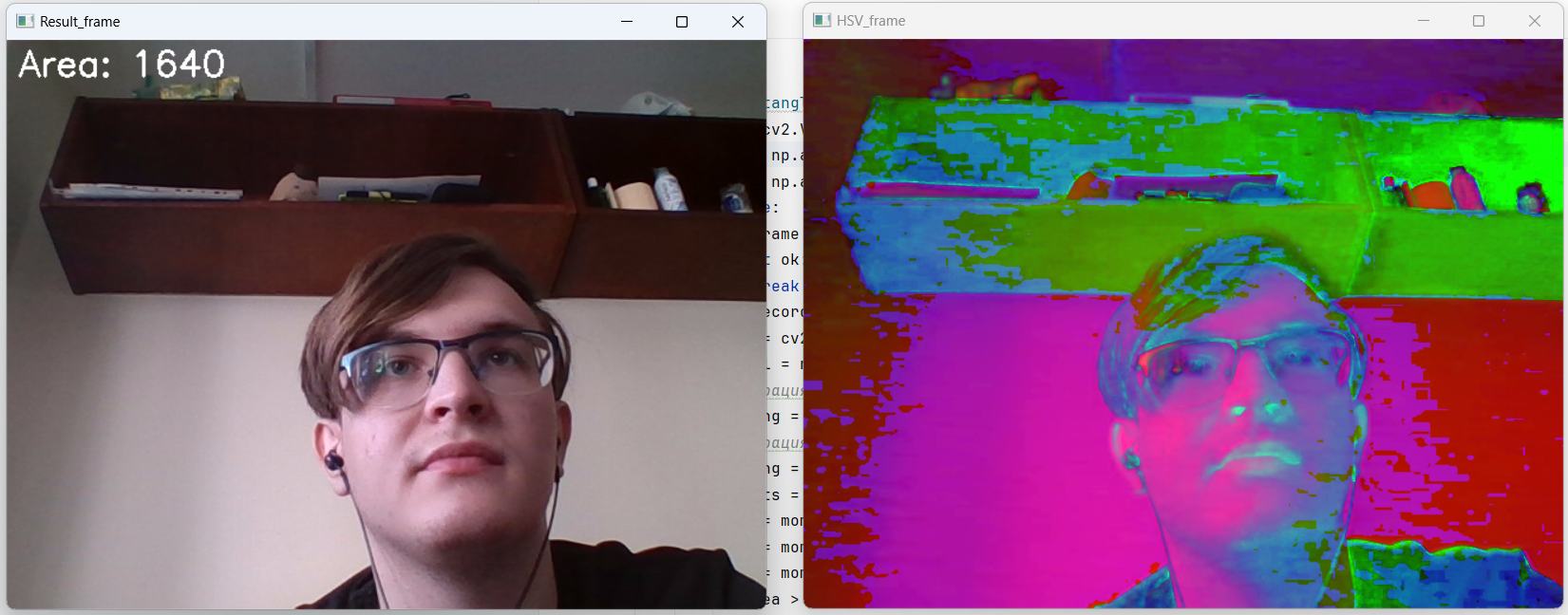
**Задание 5**

На основе анализа площади объекта найти его центр и построить черный прямоугольник вокруг объекта. Сделать так, чтобы на видео выводился полученный черный прямоугольник, причем на новом кадре новый.

Была написана функция black\_rectangle(). Используем прошлый код из 4 задания, но слегка изменив его, чтобы выделялся чёрный квадрат, когда встретился объект. Были использованы и другие моменты: m01 представляет собой сумму Y координат точек, m10 – сумма X координат точек. Для того, чтобы найти координаты центра объекта необходимо разделить m01 на m00 и m10 на m00. Центроид объекта изображения — это точка, которая представляет "центр масс" объекта. Он вычисляется на основе первых моментов изображения. Центроид используется для определения положения объекта в изображении. С помощью команды rectangle и найденного центра построим чёрный прямоугольник вокруг красного объекта.



Результат представлен на рисунке 5



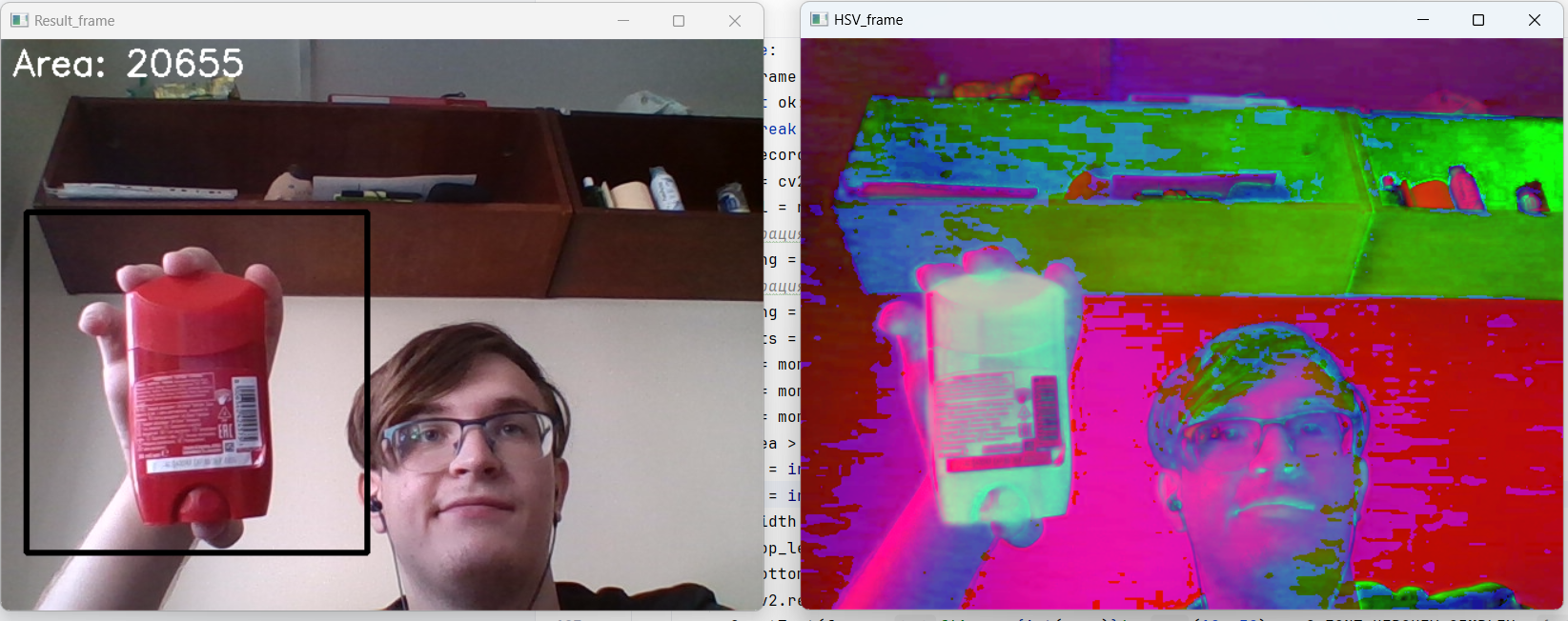


Рисунок 5 – Момент изображения с чёрным прямоугольником.

**Вывод**

Все задачи были выполнены. Был реализован трекинг красного объекта в камере, красный объект необходимо было поднести к камере, система его находит и выделяет черным прямоугольником, далее при движении красного объекта перед камерой черный прямоугольник движется за ним.