МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КУБГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**Отчет**

**по лабораторной работе №1 по курсу**

**«АЛГОРИТМЫ ЦИФРОВОЙ ОБРАБОТКИ МУЛЬТИМЕДИА»**

Работу выполнил

Студент 49 группы

Иванова В. А

Преподаватель:

Крамаренко А. А.

Краснодар

2024

**Цель работы:** реализация трекинга.

**Ход работы:**

**Задание №1**. Прочитать изображение с камеры и перевести его в формат HSV.

Используем cv2.COLOR\_BGR2HSV для перехода из стандартного формата RGB и с помощью cv2.imshow выводим на экран в окне.

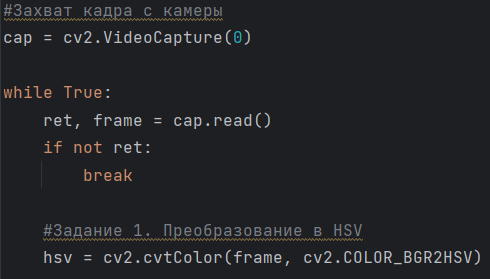
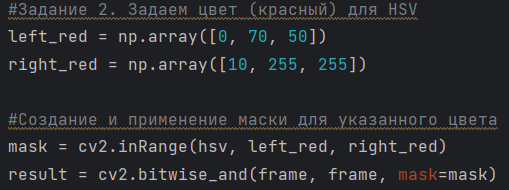
  
Рисунок 1 – перевод полученных кадров в HSV

  
Рисунок 2 – результат программы

**Задание №2.** Применить фильтрацию изображения с помощью команды inRange и оставить только красную часть, вывести получившееся изображение на экран (treshold), выбрать красный объект и протестировать параметры фильтрации, подобрав их нужного уровня.

Для нахождения цвета задаем границы определения необходимого нам цвета (от темного к светлому) через np.array. Создаем маску, где через cv2.inRange фильтруем цвет, применяем его на перевод HSV и, используя сv2.bitwise\_and (логическое умножение массивов с наложением маски) получаем необходимое выделение.

  
Рисунок 3 – задание границ цвета, создание и применение маски

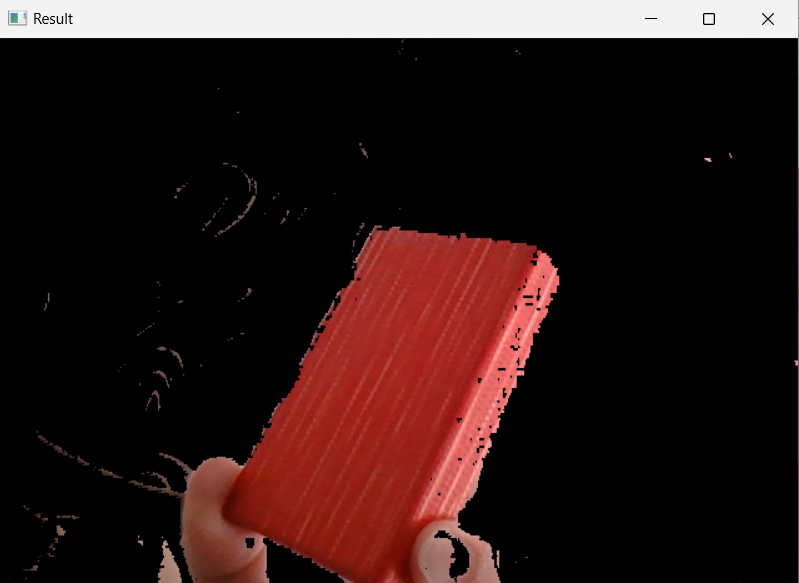
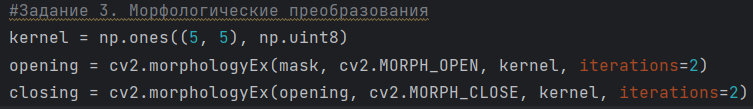
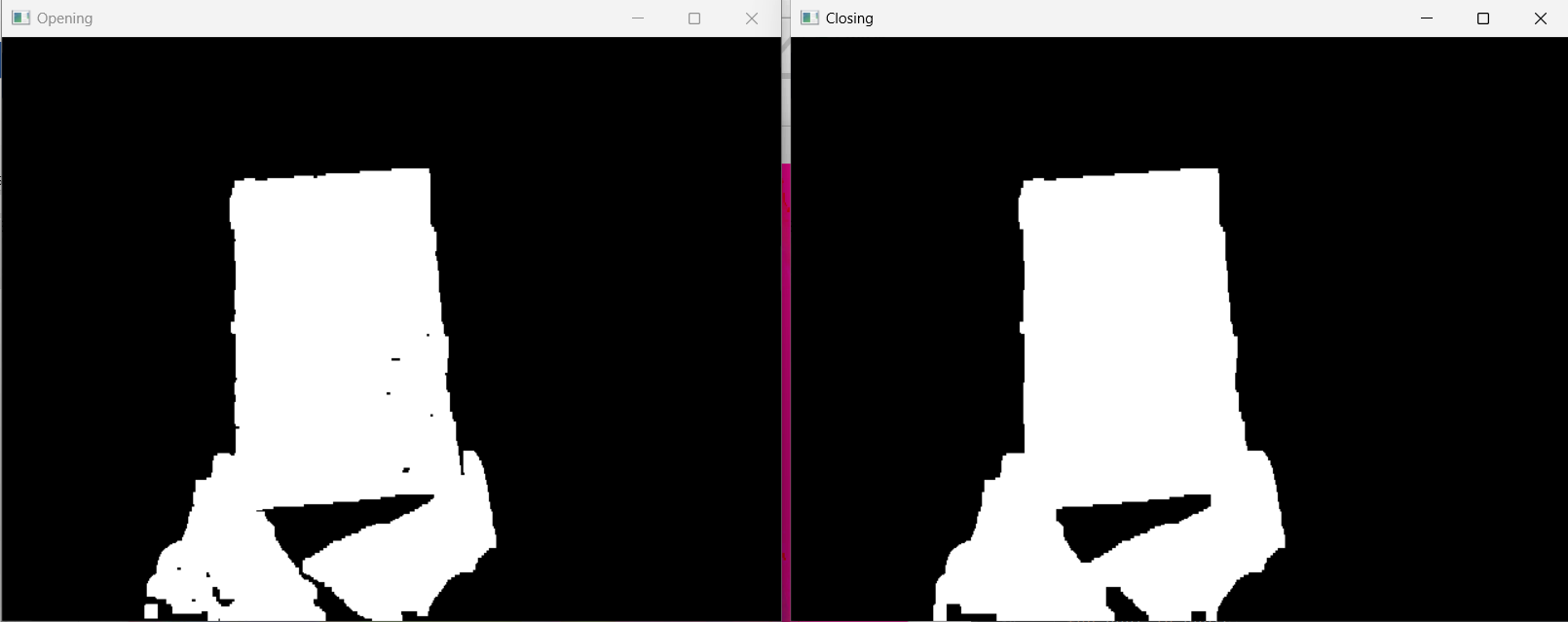


Рисунок 4 – результат работы программы

**Задание №3**. Провести морфологические преобразования (открытие и закрытие) фильтрованного изображения, вывести результаты на экран, посмотреть смысл подобного применения операций erode и dilate.

Задаем ядро (kernel) для преобразований через np.ones (создание нового массива заданного размера и типа данных, заполненный единицами). В opening используем cv2.morphologyEx с аргументами маски, cv2.MORPH\_OPEN (код операции открытия. Сначала применяется эрозия к маске, затем дилатация), ядра и двумя повторениями операции. В closing используем аналогичный ход, только на вход поступает не маска, а opening, cv2.MORPH\_CLOSE (код операции закрытия. Сначала применяется дилатация, затем эрозия).

  
Рисунок 5 – морфологические преобразования

  
Рисунок 6 – результат работы программы

**Задание №4**. Найти моменты на полученном изображении 1 первого порядка, найти площадь объекта.

**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.

Для нахождения моментов определяем контуры через cv2.findContours, где используем closing из задания №3 и указываем нахождение внешних контуров с удалением ненужных точек (cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE). Создаем для удобства копию кадра и в цикле вычисляем моменты по контурам (cv2.moments) с определением площади объекта. Обрабатываем контур, пока его площадь больше 1000 пикселей.

С помощью cv2.boundingRect определяем прямоугольник вокруг найденного контура, отрисовываем его через cv2.rectangle и над его верхней границей выводим информацию о площади объекта (cv2.putText).

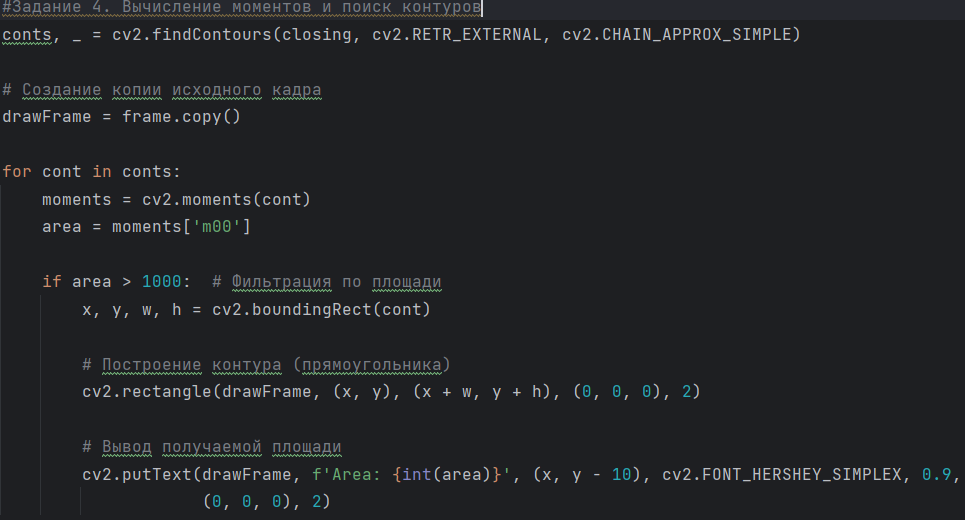
  
Рисунок 7 – определение контуров, моментов и площади объекта

  
Рисунок 8 – результат работы программы