Санкт-Петербургский политехнический университет Петра Великого
Институт машиностроения, материалов и транспорта
Высшая школа автоматизации и робототехники

Отчёт

по лабораторной работе №4

Дисциплина: Техническое зрение

Тема: Распознавание образов на изображении при помощи контурного анализа

Студент гр. 3331506/70401 Водорезов Г.И.

Преподаватель Варлашин В.В.

« »_____2020 г.

Санкт-Петербург

2020

Цель

Ознакомление со встроенными функциями для контурного анализа из библиотеки *OpenCV*, и их использование для нахождения необходимых объектов на заданных изображениях.

Задачи

1) Найти и обозначить примерный центр выделяющегося объекта на заданном изображении (рисунок 1);

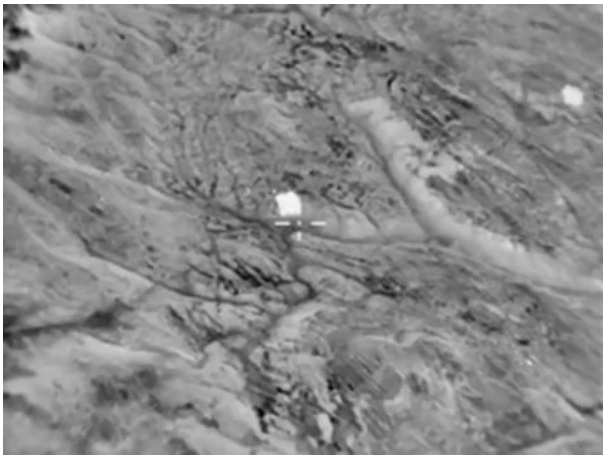

Рисунок 1 – Задание 1

2) Найти и обозначить примерный центр выделяющегося объекта на заданном изображении (рисунок 2);

Рисунок 2 – Задание 2

3.1) На каждом роботе найти его цветную верхнюю крышку и обвести контуром цвета его команды;

3.2) Найти лампу, обозначить её как-нибудь;

3.3) Для каждой команды обозначить ближайшего робота к лампе, путём рисования его центра масс;

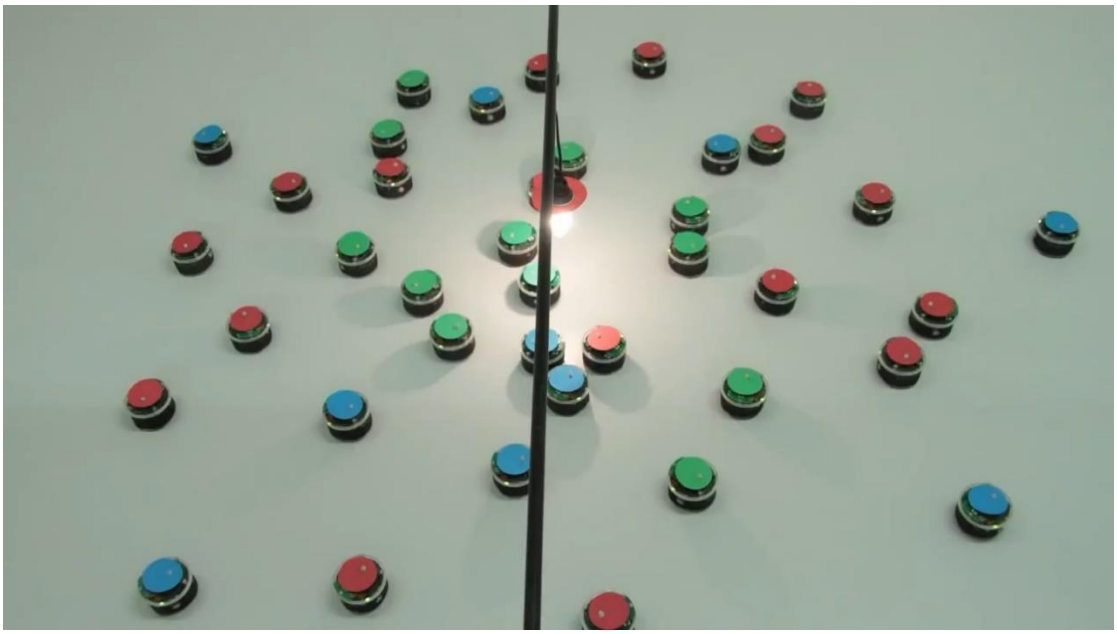

Рисунок 3– Задание 3

4) Найти и обозначить на рисунке 4 правильные и бракованные гаечные ключи с помощью заданного шаблона.

Рисунок 4 – Задание 4

Ход работы

Решение задачи 1

Последовательность решения данного задания:

1) Проводим пороговую фильтрацию с помощью функции *threshold()*. Для удаления мелких дефектов используется открывающий фильтр, который состоит из эрозии и дилатации, реализованные функциями *erode()* и *dilate()* соответственно. Результат показан на рисунке 5.

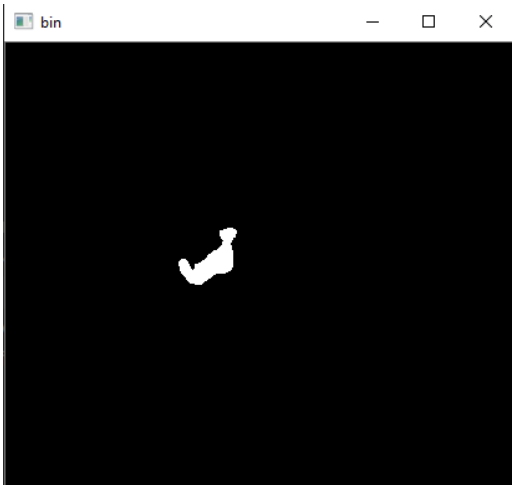

Рисунок 5 – Результат пороговой фильтрации

3) Функцией находим контур функцией *findContours()* и рисуем его функцией *drawContours()*. Для нахождения центра масс контура используем класс *Moments* и функцию *moments*. Обведем контур искомого объекта и обозначаем центр масс красной точкой на исходной изображении. Результат показан на рисунке 6.

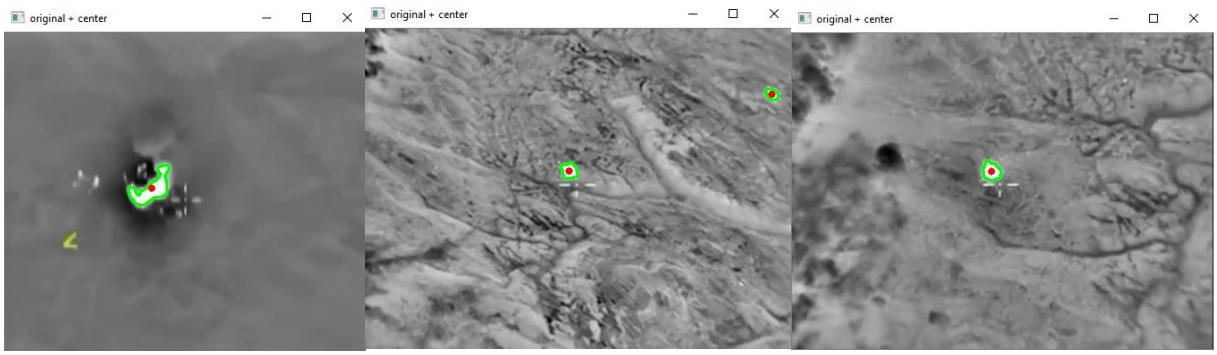

Рисунок 6 – Результат

Решение задачи 2

Последовательность решения данного задания:

1) Так как на каждом изображение моторное отделение обозначено в разных цветовых диапазонах, а также что бы не выделить лишних частей самолета был добавлен подстроечный слайдер *hsv max* (рисунок 7)

Рисунок 7 – исходное изображение и подстроечный слайдер

2) Переводим изображение из пространства BGR в цветовое пространство HSV с помощью функции *cvtColor()*. Проводим пороговую фильтрацию функцией *inRange()*.
3) Для удаления мелких дефектов проводим эрозию и дилатацию функциями *erode()* и *dilate()*. Результат показан на рисунке 8.

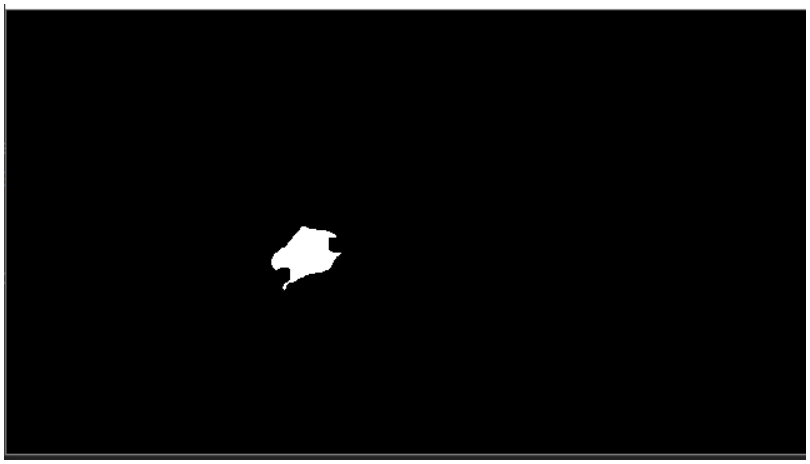

Рисунок 8 – Результат пороговой фильтрации

3) Функцией находим контуры функцией *findContours()*. Для нахождения центра масс контуров используем класс *Moments* и функцию *moments()*. Обозначаем центр масс черной точкой на исходной изображении. Результат показан на рисунке 9.

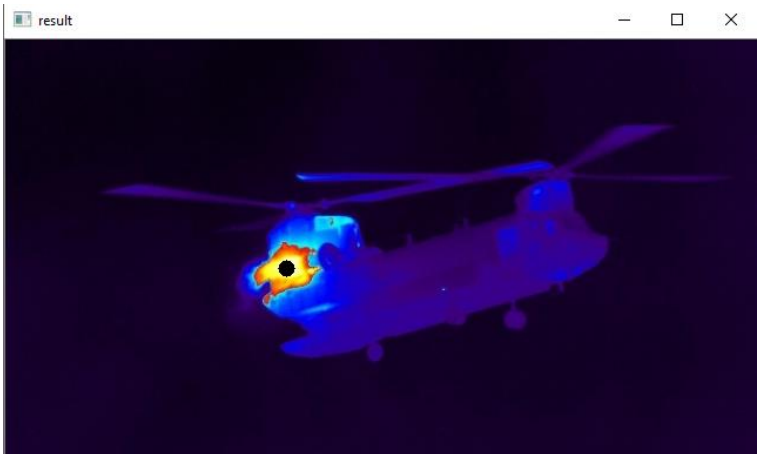

Рисунок 9 – Результат

На рисунке 10 показаны результаты работы алгоритма с другими примерами.

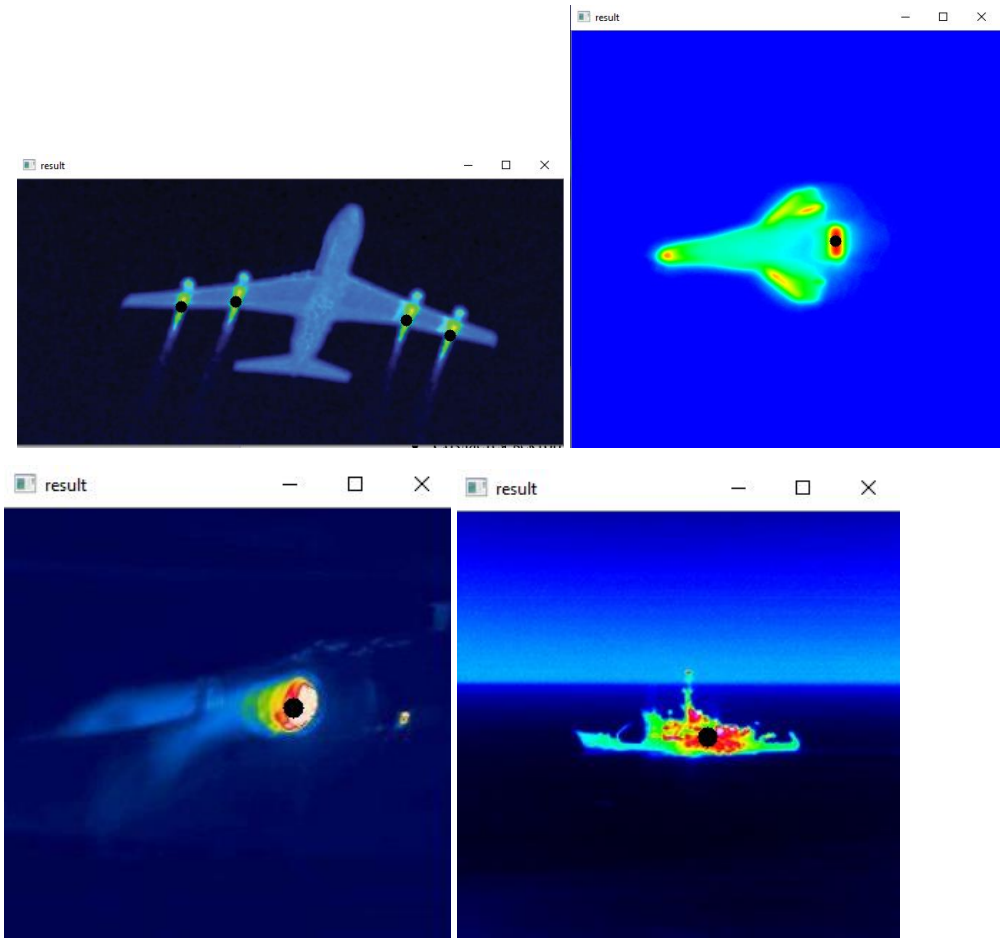

Рисунок 10 – Результаты обработки изображений

Решение задачи 3

Алгоритм решения данного задания:

1) Переводим изображение в цветовое пространство HSV с помощью функции *cvtColor()*. Проводим поиск роботов с крышками разных цветов и лампы. Для этого по аналогии с предыдущей частью используются следующие функции: *inRange(), erode(), dilate(), findContours(), moments(), elipse().*

В процессе поиска роботов красного цвете было замечено, что лампа так же обнаруживается, как и роботы, так как тоже имеет красный цвет. Для решения данной проблемы было решено нарисовать эллипс нейтрального цвета поверх лампы как на рисунке 11.

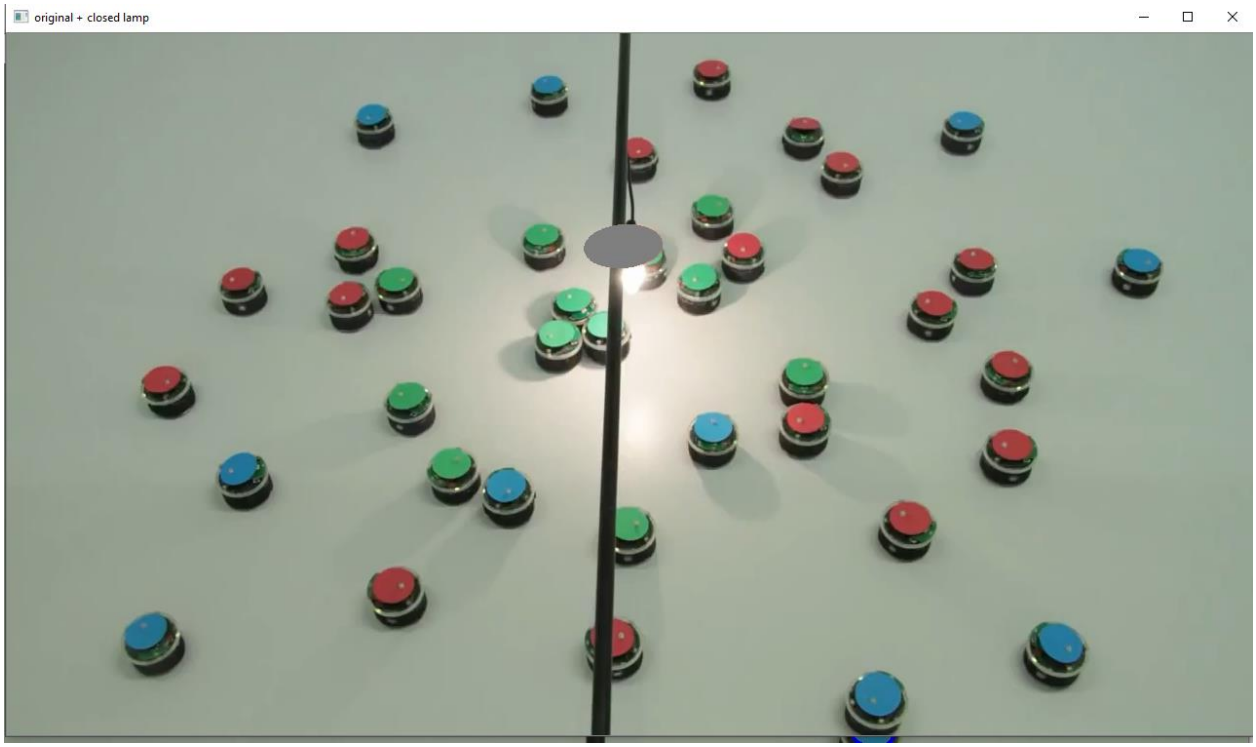

Рисунок 11 – Закрытая лампа

2) Поиск ближайшей до лампы крышки производится следующим образом:

- Создается вектор точек (координаты центров масс) всех контуров определенного цвета;
- Вычисляются расстояния между каждой точкой и лампой и находится наименьшее расстояние, реализовано функцией

findNearRobot(). Код функции на языке *Python* представлен на рисунке 12;

- Центр масс контура, который находится ближе всего к лампе, соединяется с лампой линией.

Результат работы всего алгоритма представлен на рисунке 13.

```python
# ищет ближайшего робота к лампе
def findNearestRobot(centres, lamp_centre):
    dst_min = math.sqrt((centres[0][0] - lamp_centre[0])**2 + (centres[0][1] - lamp_centre[1])**2 )
    near_center = centres[0]
    for cntr in centres:
        dst = math.sqrt((cntr[0]-lamp_centre[0])**2 + (cntr[1]-lamp_centre[1])**2)
        if dst < dst_min:
            dst_min = dst
            near_center = cntr
    return near_center
```

Рисунок 12 – Функция *findNearRobot()*

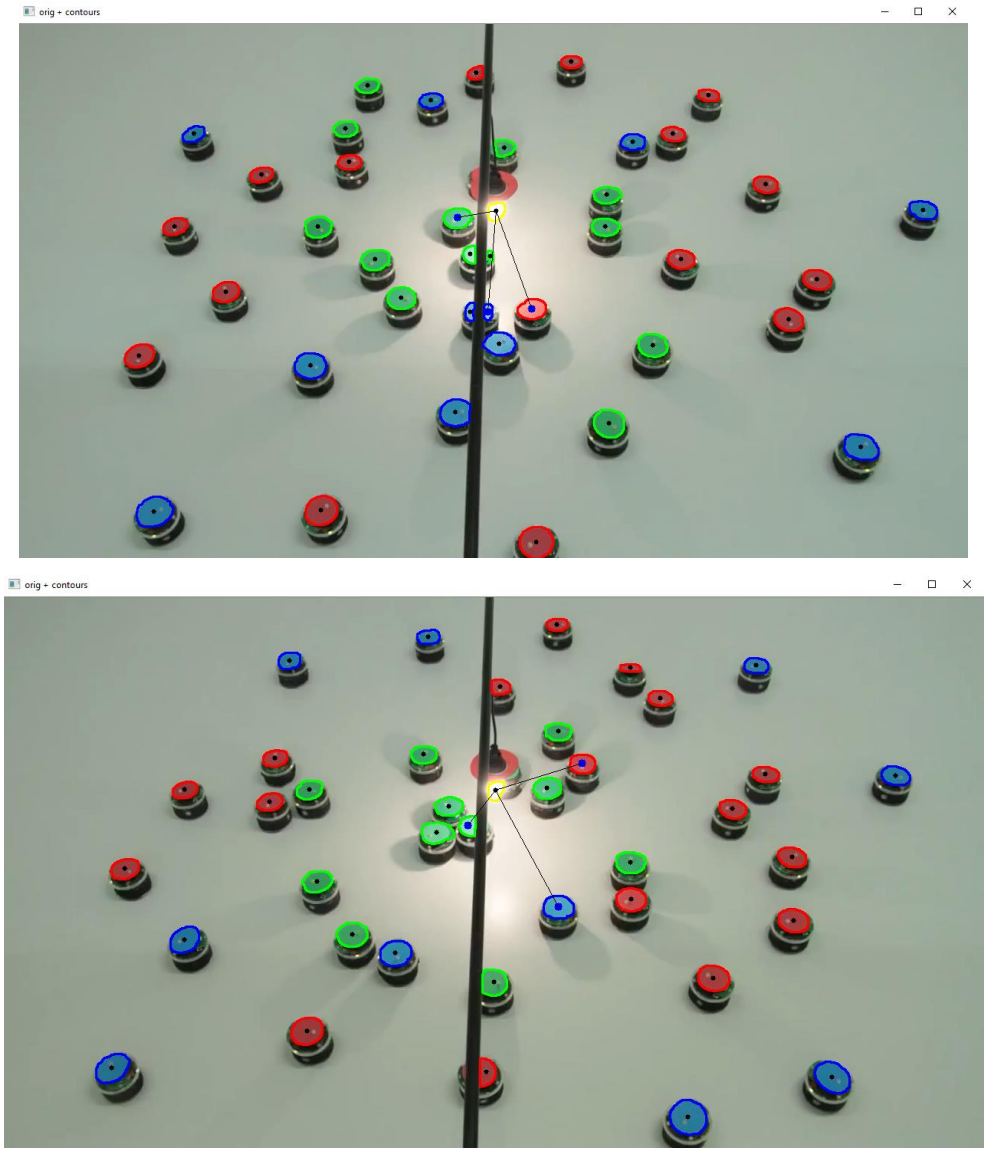

Рисунок 13 – Результат классификации роботов

Решение задачи 4

1) Сначала производим обработку шаблона:

- Пороговая фильтрация функцией *threshold()*;
- Нахождение контура и его изображение функциями *findContours()* и *drawContour()*.

2) Обработка изображения с ключами:

- Пороговая фильтрация функцией *threshold()*;
- Обработка открывающим фильтром с помощью функций *erode()* и *dilate()*;
- Нахождение контура и его изображение функциями *findContours()* и *drawContour()*.

3) Проводим сравнение контуров ключей с контуром шаблона с помощью функции *matchShapes()*.

4) Ключи, которые имеют показатель схожести меньше 1 обводим синим цветом и остальных (бракованных) красным цветом.

Результат показан на рисунке 14.

Рисунок 14 – Результат выполнения 4 задания

Вывод

В ходе работы были изучены основные функции контурного анализа из библиотеки *OpenCV*. С помощью данных функций были успешно распознаны образы объектов на заданных изображениях.