

Лабораторная работа №1. Отчет. курс «Обработка и распознавание изображений».

Пацация Александр, 317 группа

30 марта 2022 г.

1 Постановка задачи

Целью задания является разработка программы для работы с карточками игрового набора "Геометрика". С помощью точечных и пространственных преобразований требуется:

1. Определить количество карточек на изображении
2. Определить является ли фигура многоугольником или фигурой с гладкой границей.
3. Определить количество вершин многоугольников
4. Проверить факт выпуклости многоугольников

2 Описание данных

Входные данные – трёхканальные фотографии карточек набора "Геометрика" в формате JPG. На рис. 1 приведены примеры входных изображений.



(a) IMG_1.jpg



(b) IMG_4.jpg



(c) IMG_10.jpg

Рис. 1: Примеры входных изображений

Как можно видеть, есть несколько типов изображений:

1. Изображения с однородным фоном, карточки не накладываются.
2. Изображения с однородным фоном, карточки накладываются.
3. Изображения с пестрым фоном, карточки не накладываются.

3 Метод решения

Решение состоит из двух этапов:

1. Сегментация фигур.
2. Анализ фигур: определение количества вершин для каждой фигуры и проверка на выпуклость.

3.1 Сегментация фигур

Для начала стоит избавиться от большинства шумовых объектов на карточках (транспортиры, линейки и т.д.). Способ, который будет продемонстрирован далее, позволяет сразу же избавиться не только от ненужных объектов, но и от пестрого фона. Первым делом, переведем изображения из модели RGB в HSV и рассмотрим первый канал – Ние, отвечающий за цветовой тон.

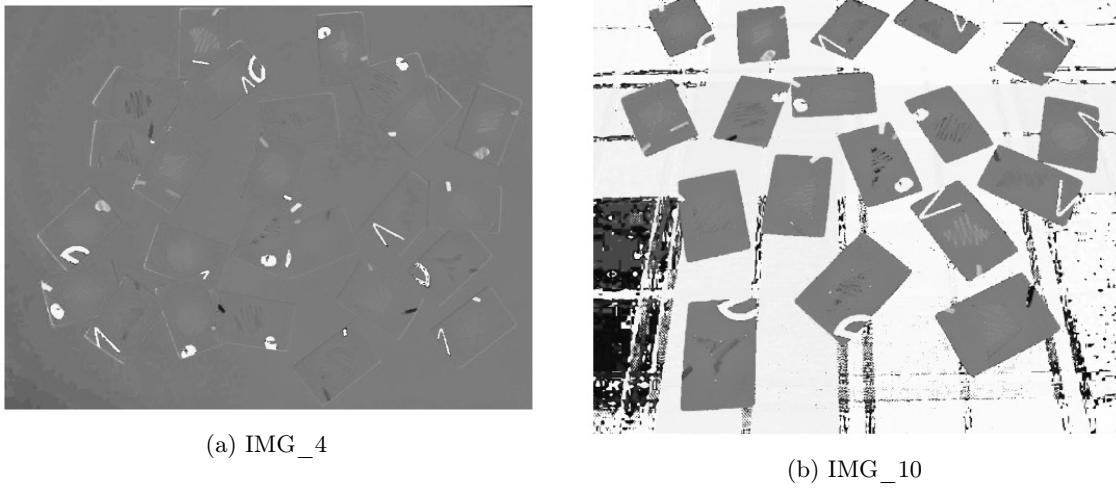


Рис. 2: Изображения в канале Ние

Как можно видеть на рис. 2 данный формат не только способен отделить лишние объекты, но и отделить карточки от пестрого фона. Чтобы окончательно в этом убедиться можно взглянуть на гистограмму изображений в этом канале рис. 3.

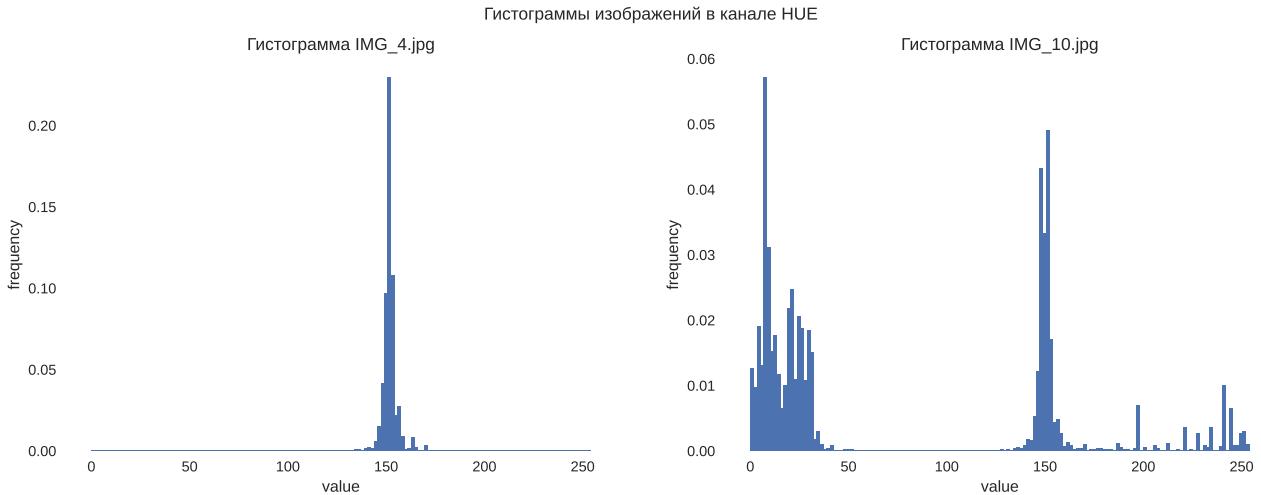
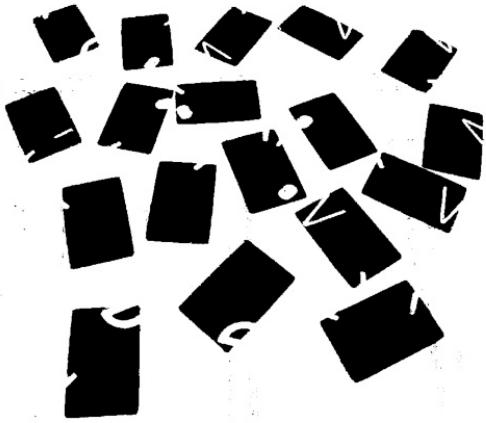


Рис. 3: Гистограммы в канале Ние

Диапазон значений [125, 170] соответствует карточкам и фону, в случае однородного фона и только карточкам, в случае пестрого фона (проверено экспериментально). Бинаризовав по диапазону [125, 170] и заполнив небольшие отверстия, получим маски представленные на рис. 4.



(a) IMG_4



(b) IMG_10

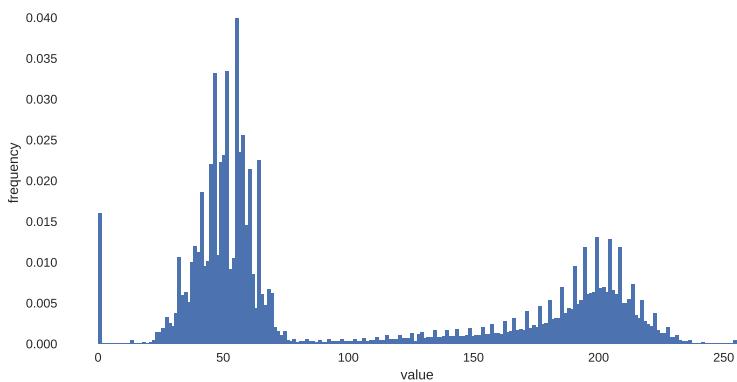
Рис. 4: Маски удаляющие фон и ненужные объекты

Проблема с пестрым фоном была решена вместе с проблемой шумовых объектов, однако если на вход подается картинка с однородным фоном, то надо уметь убирать и его. Снова используем модель HSV, на этот раз канал Saturaion. Применим полученную ранее маску, чтобы немного очистить распределение. Гистограмма и соответствующее изображение приведены на рис. 5



(a) IMG_4 (Saturation + маска)

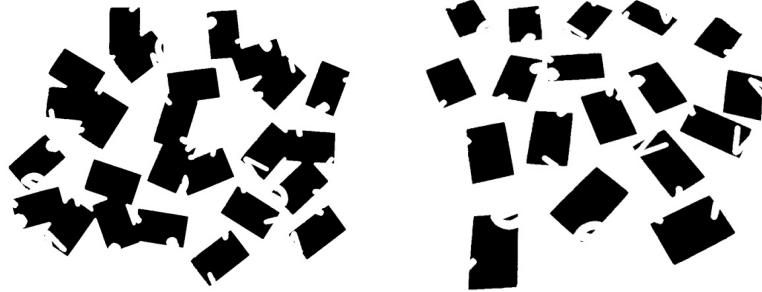
Гистограмма IMG_4 в канале Saturation после применения маски



(b) гистограмма IMG4

Рис. 5: IMG4 в канале Saturation

Распределения фона и карточек опять разделяются. Чтобы оптимально бинаризовать изображение используем алгоритм Оцу. Далее заполним дыры и применим операцию эрозии, чтобы отойти от границ и захватить с собой тени шумовых объектов. Результат приведен на рис. 6

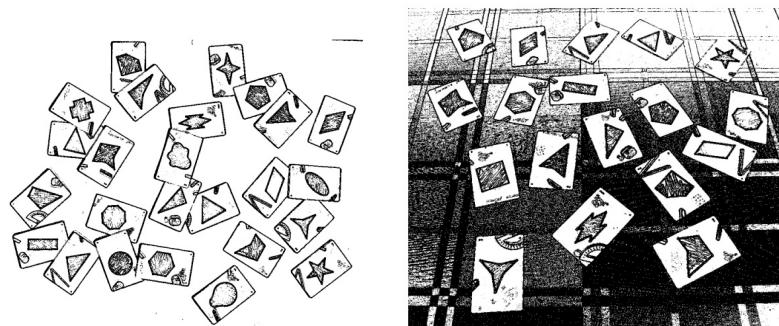


(a) финальная маска IMG_4

(b) финальная маска IMG_10

Рис. 6: IMG4 в канале Saturation

Как видно, из-за того что мы применили предыдущую маску, текущий шаг никак не испортил маску изображения с пестрым фоном (IMG_10). Финальная маска получена, теперь надо найти границы объектов. Для этого применим фильтр Собеля, предварительно переведя исходное изображение в серое. Далее бинаризуем изображение по некоторому низкому нижнему порогу (рис. 7).

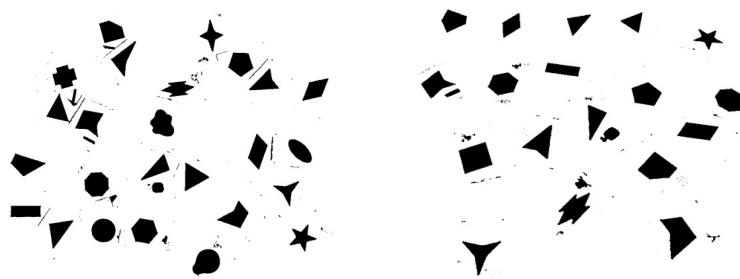


(a) бинаризованные границы IMG_4

(b) бинаризованные границы
IMG_10

Рис. 7: границы объектов

Теперь применим финальную маску и заполним дыры, чтобы залить интересующие нас фигуры. Результат можно увидеть на рис. 8.

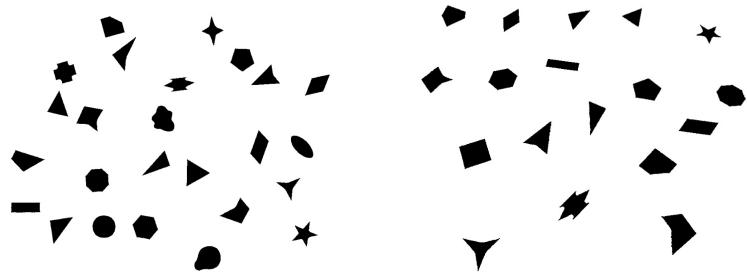


(a) фигуры IMG_4

(b) фигуры IMG_10

Рис. 8: фигуры перед очисткой от шума

Можно заметить, что все шумовые объекты заметно меньше фигур, поэтому от них можно избавиться, убрав объекты меньше определенного порога, предварительно применив операцию эрозии с небольшим примитивом, чтобы отделить слившиеся шумовые объекты и фигуры, если таковые имеются. Видно (рис. 9), что фигуры практически идеально сегментируются



(a) фигуры IMG_4

(b) фигуры IMG_10

Рис. 9: сегментированные фигуры

3.2 Анализ фигур

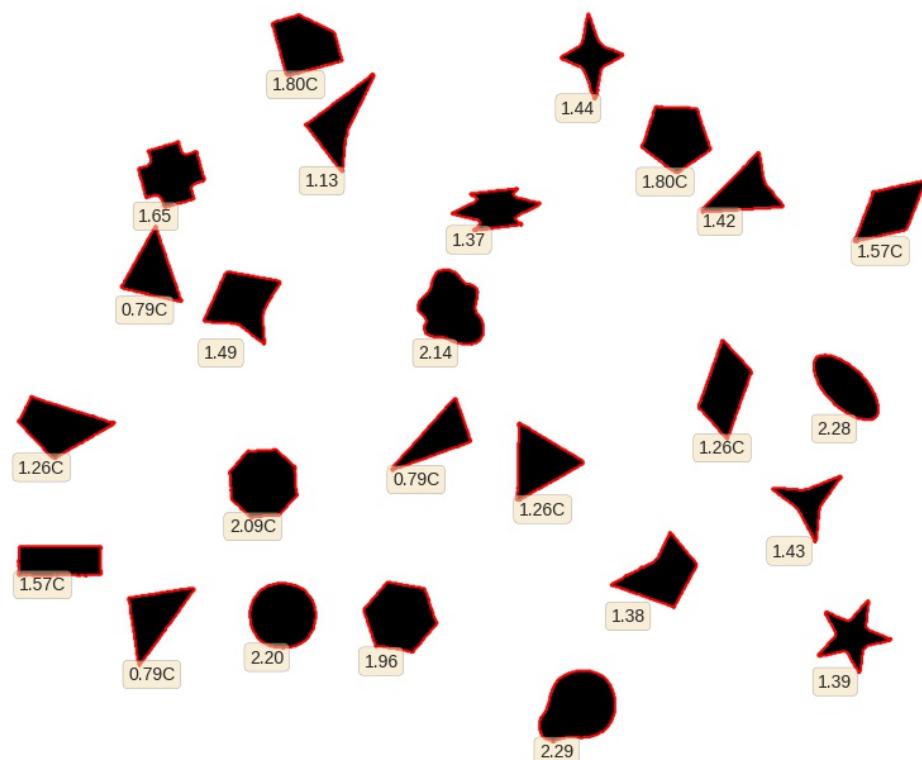


Рис. 10: Аппроксимация объектов IMG_4 полигонами с низким порогом отклонения. Каждый многоугольник помечен суммой углов в радианах

После сегментации фигур требуется провести их анализ. Для начала построим контуры найденных фигур, используя алгоритм "марширующих квадратов" и последующей линейной интерполяции границ. Далее с помощью алгоритма Дугласа-Пекера построим аппроксимирующие многоугольники с двумя порогами отклонения от исходных контуров: достаточно большим и низким. Первое представление позволяет получить точную аппроксимацию многоугольников, второе позволит понять какие фигуры многоугольниками не являются.

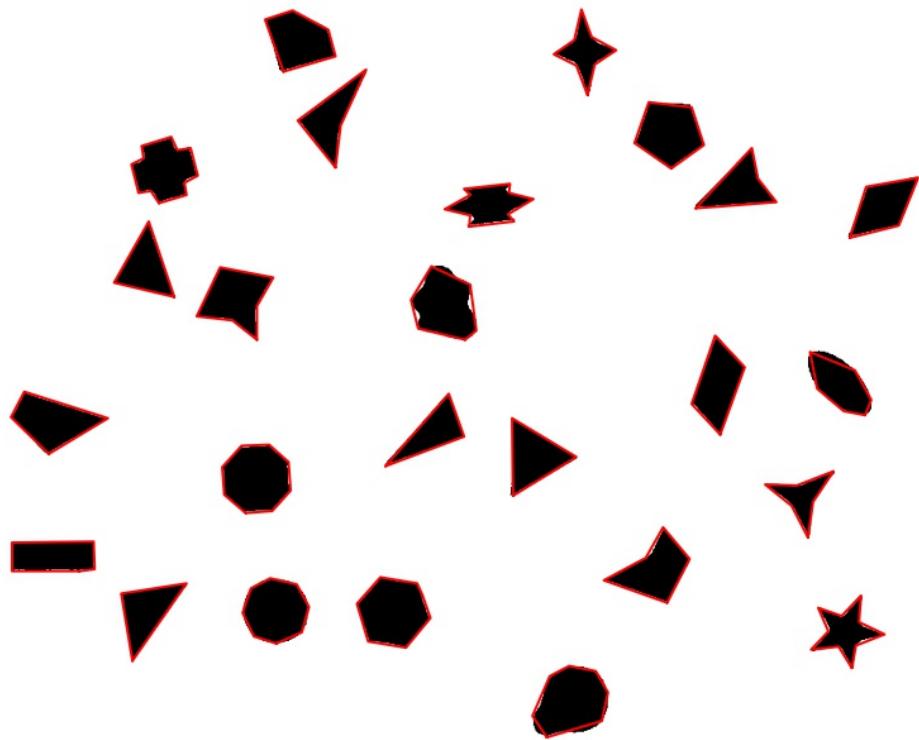


Рис. 11: Аппроксимация объектов IMG_4 полигонами с высоким порогом отклонения

Возьмем второе представление и для каждого многоугольника посчитаем коэффициент: среднее значение угла аппроксимирующего многоугольника в радианах. Результат изображен на (рис. 10). Видно, что фигуры с гладкой границей имеют превосходящие значения среднего угла по сравнению с многоугольниками: для многоугольников он не превышает 2.09, в то время как у фигур с гладкой границей он не меньше 2.2. Это происходит из-за того, что аппроксимация гладких фигур содержит достаточно много ребер между которыми достаточно большие углы. Выберем порог значения коэффициента в 2.1. Все фигуры с меньшим значением будем относить к многоугольникам, с большим – к гладким фигурам. Сразу стоит отметить, что выбранный критерий не будет хорошо работать, если на изображении есть выпуклые многоугольники с большим количеством ребер, например правильный восьмиугольник имеет большое значение 2.09.

Теперь рассмотрим аппроксимацию с высоким порогом отклонения (рис. 11). Иногда в аппроксимации попадаются лишние вершины. Их два вида:

1. Вершины угол при которых близок к π .
2. Вершины, находящиеся очень близко в других вершинам.

После избавления от подобных вершин, для каждого многоугольника мы получаем корректное количество вершин.

Проверка на выпуклость осуществляется следующим образом: по вершинам аппроксимирующего многоугольника строится выпуклая оболочка, если она содержит не все вершины многоугольника значит многоугольник не выпуклый. Итоговый выход программы представлен на (рис. 12)

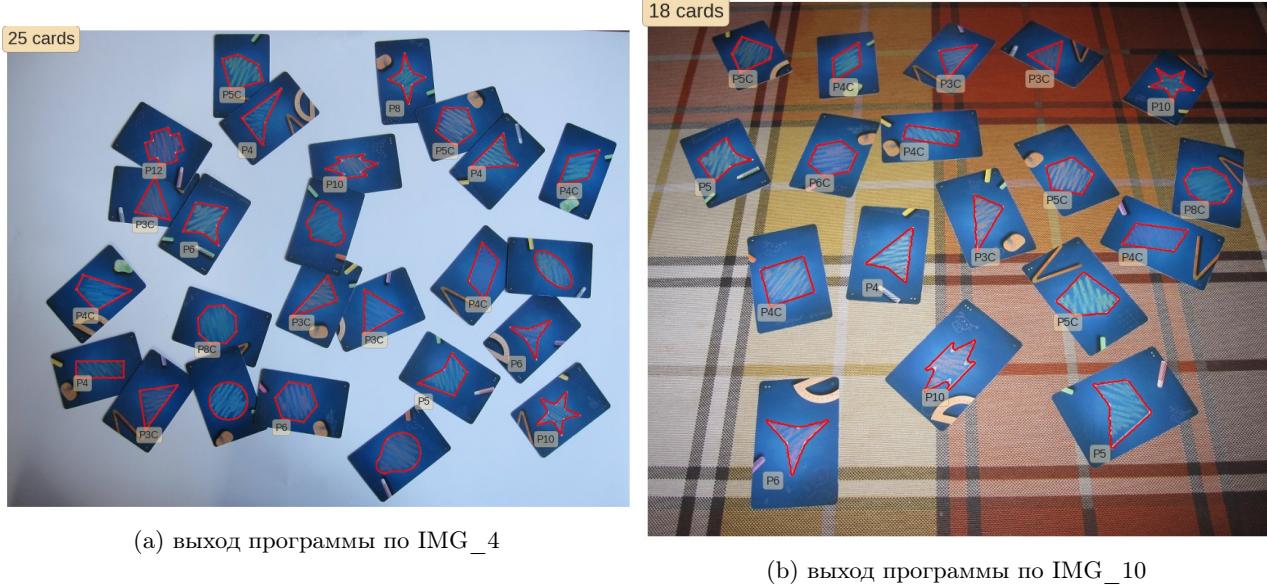


Рис. 12: Итоговый выход программы

4 Программная реализация

Программа реализована на языке Python с использованием следующих библиотек:

1. skimage, scipy.ndimage для обработки изображений
2. numpy для матричных вычислений
3. matplotlib для отрисовки и сохранения обработанных изображений

Запуск программы с помощью командной строки:

```
python geometrica_recognition.py input_image output_directory
```

`input_image` – путь к входному изображению. `output_directory` – путь к выходной директории. Если `output_directory` не указан, по умолчанию выходное изображение сохраняется в текущей директории. Программа обрабатывает изображение, помещает выходной файл в указанную директорию и выводит его на экран. Формат выходного файла – JPG. Также является возможным импортирование функций из `geometrica_recognition.py` для экспериментов с параметрами.

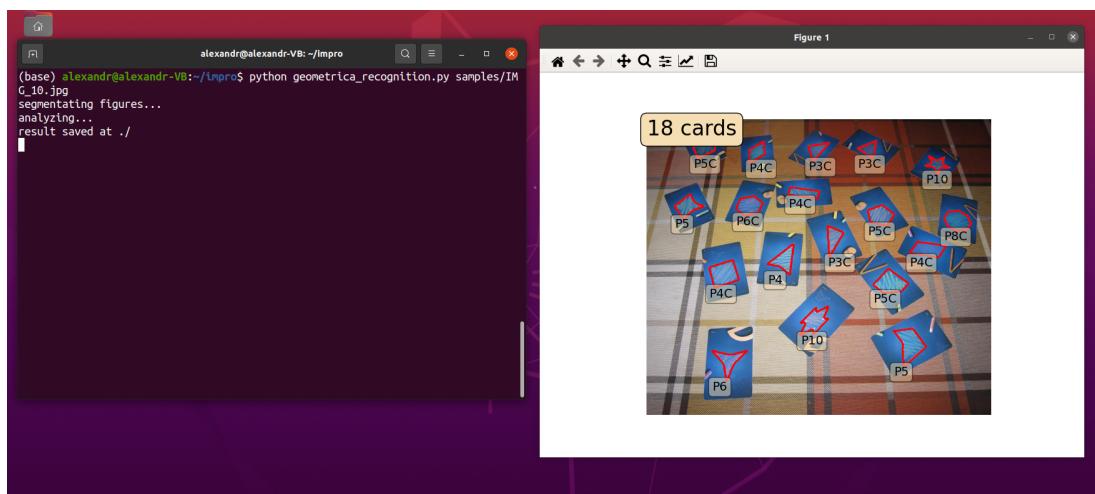


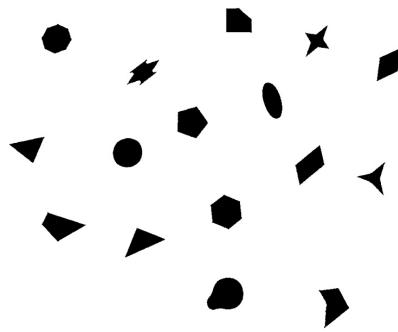
Рис. 13: Пример работы программы

5 Эксперименты

В предыдущих разделах подробно рассматривалась работа программы для IMG_4.jpg и IMG_10.jpg. Посмотрим как алгоритм работает с другими изображениями. Для начала проведем эксперименты с сегментацией.



(a) IMG_1

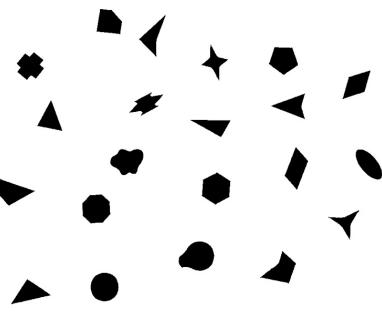


(b) сегментированный IMG_1

Рис. 14: Сегментация IMG_1



(a) IMG_2

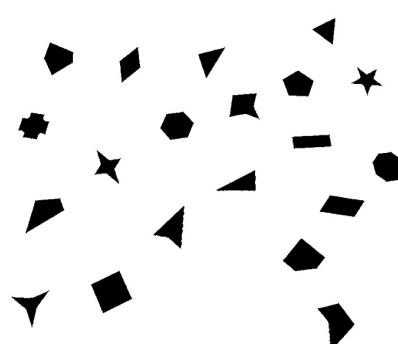


(b) сегментированный IMG_2

Рис. 15: Сегментация IMG_2



(a) IMG_9



(b) сегментированный IMG_9

Рис. 16: Сегментация IMG_9

Как видно, проблемы возникают когда карточка засвеченна (рис. 16), границы засвеченного элемента не выделяются. Теперь рассмотрим выходы программы и оценим качество обработки.



Рис. 17: Выход программы по IMG 6



Рис. 18: Выход программы по IMG 7

Программа хорошо обрабатывает многоугольники даже с большим количеством вершин, однако, как было отмечено выше, проблемы могут возникнуть если на карточке будет изображен выпуклый многоугольник с большим количеством вершин. Подсчет количества карточек, подсчет количества вершин и проверка на выпуклость осуществляются корректно для всех изображений.

6 Выводы

По итогам лабораторной работы была разработана программа для обработки карточек набора "Геометрика". Программа сегментирует и анализирует фигуры на карточках. Сегментация проводится хорошо для всех карточек, кроме засвеченных. Используемый критерий разделения многоугольников и гладких фигур хорошо работает на предложенных изображениях, однако допускается возможность ошибки на выпуклых многоугольниках с большим количеством вершин. Поскольку в процессе обработки не происходит обучения, алгоритм не универсален (например, не устойчив к масштабу) и при поступлении новых данных может потребоваться перенастройка параметров.