

Отчет по лабораторной работе №1

курс «Обработка и распознавание изображений»
кафедра ММП ВМК МГУ

Михеев Борис, 317 группа

29 марта 2022 г.

Постановка задачи

Цель работы - разработать и реализовать программу для работы с изображениями карточек игрового набора «Геометрика». Требования к программе:

- Ввод и отображение на экране изображений.
- Сегментация и подсчет числа карточек.
- Сегментация фигур на карточках и определение их типа - многоугольник или фигура с гладкой границей. Для многоугольников программа должна определить, является ли фигура выпуклой, и провести подсчет вершин.
- Выходом программы должно являться исходное изображение с нанесенной разметкой маркерами вида PnC или Pn : P - маркер того, что фигура - многоугольник, n - число его вершин, C - маркер того, что многоугольник является выпуклым. Также требуется нанести номер карточки, их порядок не определен. Маркеры должны находиться на изображениях соответствующих карточек в любом месте в границах карточки.

Перечисленные задачи ставятся в двух вариантах: с белым и пестрым фоном, с изолированными и перекрывающимися карточками.

Описание данных

Входные данные - снимки карточек игрового набора «Геометрика» с рисунками различных геометрических фигур. Изображения имеют формат jpg, являются трехканальными изображениями с интенсивностями пикселей от 0 до 255. Все изображения имеют различную ширину и высоту в пикселях. Также изображения отличаются по четкости, фону и освещению. В простейшем случае изображение четкое, с равномерным освещением и белым фоном. В более сложных случаях изображения могут быть несколько размытыми, с неравномерным освещением или могут иметь пестрый фон - цветной орнамент с горизонтальными и вертикальными полосами. Также карточки на снимке могут быть как изолированными, так и накладывающимися друг на друга. Предполагается, что разрабатываемая программа должна адекватно работать с такими особенностями входных изображений.



Рис. 1: Примеры входных изображений, предоставленных для отладки и тестирования. Первое изображение - простейший случай, белый фон, равномерное освещение, карточки изолированы. Второе изображение - более сложный вариант, присутствуют перекрытия. Третье изображение - наиболее высокая сложность, присутствует неравномерное освещение и фон с дополнительными элементами.

Описание метода решения

Решение задачи разделяется на следующие основные этапы. Сначала происходит сегментация самих прямоугольных карточек, далее происходит сегментация изображенных на них фигур. После строятся контуры выделенных фигур, затем они приближаются полигонами. После происходит определение, является ли фигура многоугольником, и в таком случае далее определяется его выпуклость и подсчет вершин.

Рассмотрим процесс решения более детально. Исходное трехканальное изображение переводится в одноканальное. Программа позволяет произвести перевод различным способом, выделяя различные каналы в модели RGB и HSV на усмотрение пользователя.



(a) Исходное изображение



(b) Одноканальное изображение с выделенным красным каналом

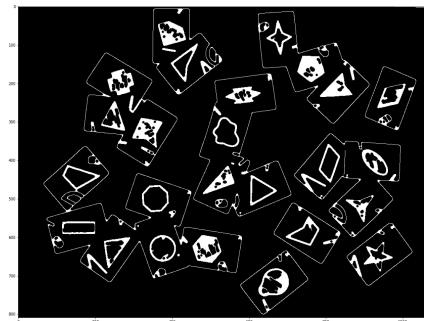
Рис. 2: Перевод исходного изображения в одноканальное.

Далее к полученному одноканальному изображению применяется медианный фильтр для снижения уровня шума. Затем для обнаружения границ на полученном изображении применяется детектор границ Кэнни.

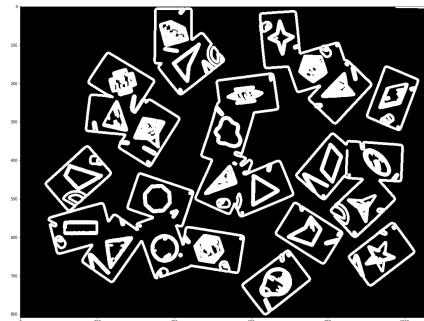


Рис. 3: Выделение границ с помощью детектора Кэнни.

Дальнейшей целью является выделение самих карточек, их прямоугольных очертаний. Будем выделять их с использованием морфологических операций. Применим последовательно операции замыкания, дилатации, заполнения областей и эрозии. В качестве примитивов в данных операциях в программе используется диск. В результате получаем заполненные контуры карточек.

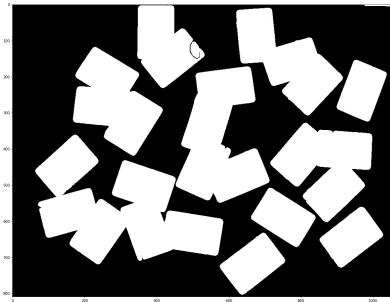


(a) Замыкание

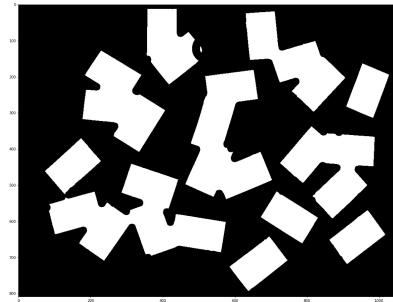


(b) Дилатация

Рис. 4: Применение морфологических операций для сегментации карточек, замыкание и дилатация.



(a) Заполнение областей



(b) Эрозия

Рис. 5: Применение морфологических операций для сегментации карточек, заполнение областей и эрозия.

Стоит отметить, что на карточках помимо интересующих геометрических фигур присутствуют сторонние объекты, сравнимые по размеру и очертаниям с фигурами: изображения ластиков, карандашей, угольников, транспортиров и т. д. Их присутствие и геометрия затрудняет дальнейшую сегментацию фигур. Но они находятся по краям карточек, и в некоторых случаях их цвет близок к цвету фона, поэтому эрозия с диском достаточно большого радиуса в качестве примитива позволит бороться с этим. Как можно заметить на Рис.5(b), на полученной в итоге маске «вырезаны» некоторые фрагменты по краям, отвечающие за сторонние объекты. При этом размеры карточек на маске меньше размеров исходных карточек. Все это в совокупности поможет упростить дальнейшую сегментацию фигур, так как полученную маску сегментации карточек мы применим к одноканальному изображению, полученному по исходному, для выделения внутренних областей карточек. Размер сторонних объектов в итоге станет меньше, и их границы, вероятно, станут разрывными.

Далее к выделенному фрагменту одноканального изображения снова применим детектор границ Кэнни.

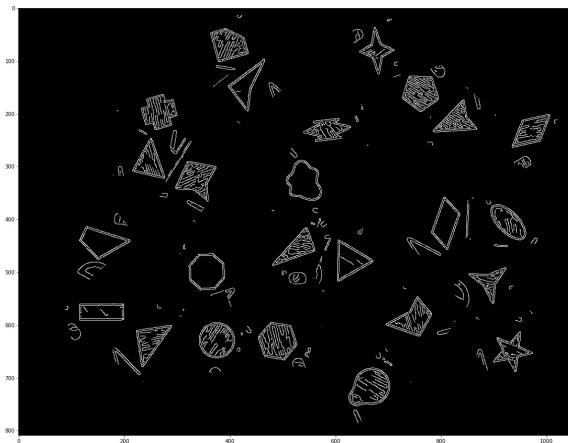
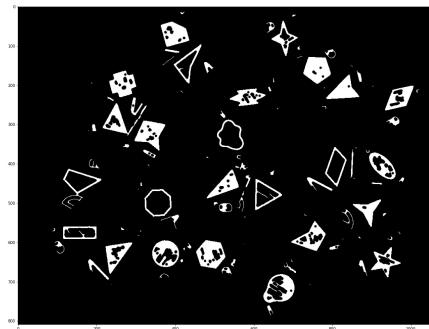


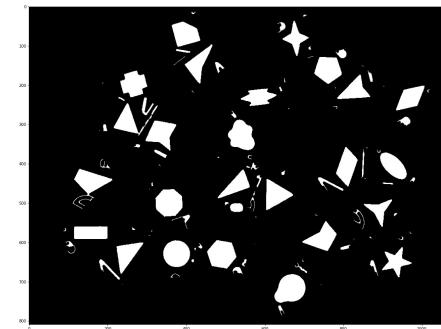
Рис. 6: Выделение границ фигур во внутренних областях карточек с помощью детектора границ Кэнни.

Как можно увидеть на Рис.6, относительно сильная эрозия при сегментации карточек действительно помогла сократить число и размер сторонних объектов и сделать их границы разрывными в некоторых случаях, что упрощает дальнейшую сегментацию фигур.

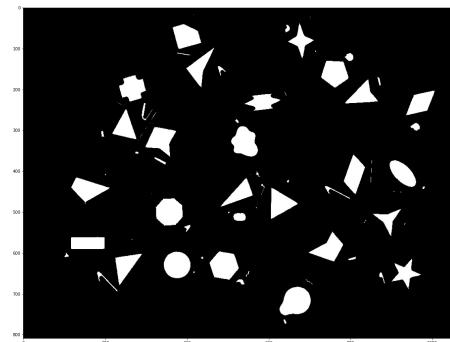
Затем аналогично используется последовательность различных морфологических операций: замыкание, заполнение областей и эрозия. Стоит отметить, что на данном этапе в отличие от сегментации карточек нам важно как можно лучше сохранить контуры и углы фигур для их дальнейшего анализа. Поэтому целесообразно попробовать использовать в качестве примитива эрозии небольшие многоугольники, в том числе и со сторонами «под наклоном», например, ромб, восьмиугольник и т. д. Среди фигур на карточках набора довольно много многоугольников с острыми углами, и также сами карточки могут быть повернуты относительно вертикали и горизонтали. В силу этого данная идея может быть полезной. В любом случае программа позволяет выбирать использующийся в эрозии примитив на усмотрение пользователя.



(a) Замыкание



(b) Заполнение областей



(c) Эрозия

Рис. 7: Применение морфологических операций для сегментации фигур: замыкание, заполнение областей и эрозия.

После проведения морфологических операций остается много мелких остаточных объектов. Так как их размер значительно меньше размера фигур, то можем просто провести удаление всех объектов меньше некоторого фиксированного размера.

В итоге получаем маску отсегментированных фигур, изображенных на карточках. Так как на каждой карточке может быть только одна фигура, то для подсчета карточек достаточно посчитать число отсегментированных фигур, т. е. число связных областей на изображении.

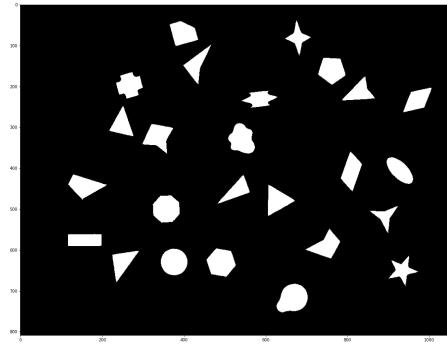
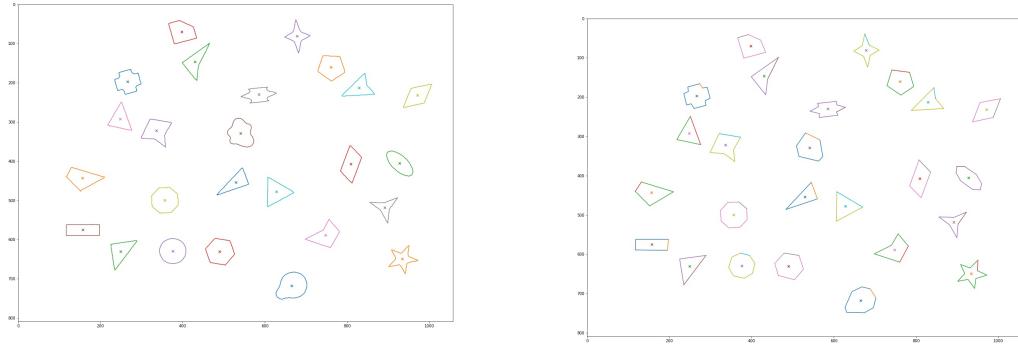


Рис. 8: Удаление мелких остаточных фрагментов.

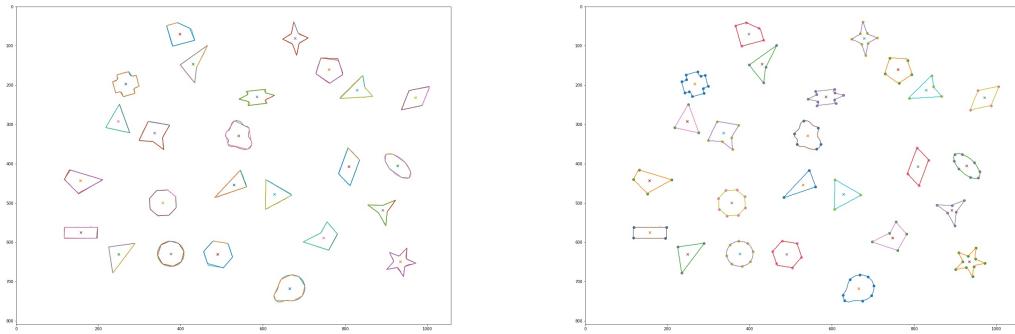
На следующем шаге требуется проанализировать фигуры. Для этого необходимо проанализировать их контуры. Построим контуры областей на изображении, полученном в ходе сегментации фигур. Но все же сегментация дает не идеально точный результат, и, соответственно, построенные по ее результатам контуры также не идеальны, хотя достаточно точно приближают рассматриваемые фигуры. Также сложно по контурам проанализировать гладкость или число углов фигуры, так как они все же содержат некоторые погрешности и учитывают погрешности сегментации. Поэтому поступим следующим образом. Каждый из полученных контуров будем приближать полигоном с не очень большой точностью, в предположении, что в случае многоугольника этот аппроксимирующий полигон достаточно хорошо приблизит фигуру, и его вершины будут очень близки к истинным вершинам фигуры.



(a) Контуры, построенные по результатам сегментации

(b) Приближающие контуры полигонами

Рис. 9: Построение контуров фигур и их приближение полигонами.



(a) Контуры фигур и их приближения

(b) Контуры фигур и вершины их приближений

Рис. 10: Совмещение контуров фигур и их приближений.

Действительно, аппроксимирующие полигоны хорошо приближают фигуры, а для многоугольников число их вершин совпадает с числом вершин исходной фигуры, и их координаты очень близки. Таким образом, чтобы посчитать число вершин в случае многоугольника, достаточно посчитать число вершин в соответствующем полигоне. Определение выпуклости контуров может быть легко совершено при помощи соответствующих функций из различных доступных библиотек, например, opencv.

Остается достаточно нетривиальный вопрос определения гладкости контура. Вообще говоря, контуры, построенные по маске сегментации фигур, неточны, состоят из множества точек и могут быть «угловатыми» во множестве мест. Поэтому сложно определить для точного контура его гладкость, например, по числу его точек или его углам. В случае менее точно приближения гладкая кривая может быть приближена, например, восьмиугольником, и с высокой вероятностью будет классифицирована как многоугольник. Вообще говоря, можно сравнивать исходный контур и полигон-приближение (вернее, его границу), например, по минимальному расстоянию между кривыми, разности площадей внутри контуров, различным метрикам вроде IoU и т. д. Однако в ходе разработки и экспериментов ни один из перечисленных способов не дал удовлетворительного результата. Детекторы углов также давали крайне посредственный результат. Поэтому была использована следующая эвристика. Рассматривались аппроксимирующие полигоны контуров и их углы, и если контур выпуклый и медианное значение в наборе его углов больше или равно некоторому порогу, то кривую можно считать гладкой, иначе это многоугольник. Здесь используется предположение, что в случае гладкой кривой углы между смежными сторонами полигона будут достаточно большими. Условие про выпуклость было взято потому, что, во первых, в ходе экспериментов гладкие фигуры из набора «Геометрики» приближались всегда выпуклыми контурами, что достаточно логично, а во вторых, в наборе есть фигуры с множеством тупых и развернутых углов, но они в подавляющем большинстве невыпуклые («звезды» различной формы и т. д.). Медианное значение берется, т. к. оно показало наилучший результат. Среднее значение угла брать не совсем правильно, ведь, например, круг может приблизиться 8-ми угольником, сумма углов в выпуклом n-угольнике - фиксированная величина, и тогда средний угол для такого приближения и правильного 8-миугольника будет одинаковым. Минимальное значение угла также не даст хороших результатов, т. к. в случае гладкой, но сложной фигуры это может быть довольно небольшое значение, в т. ч. и острый угол в общем случае (овал или грушевидная форма и т. д.). Аналогично с максимальным значением угла, порой и у правильных многоугольников оно может оказаться больше, чем, к примеру, у приближения овала или круга. Понятно, что такая эвристика крайне неточна и нестабильна. Но т. к. программа позволяет варьировать значения различных параметров, в т. ч. и точность приближения контуров полигонами и т. д., то в зависимости от конкретного входного изображения варьированием соответствующих

параметров можно добиться наиболее приемлемого возможного результата. Возможно, для более качественного определения гладкости можно использовать более сложные методы и характеристики контуров, проводить сравнение точных контуров с различными, но близкими уровнями точности (хотя в силу малости различий получаемых линий в таком случае анализ затруднен), или же варьировать параметры морфологических операций для получения как можно более точной границы и контуров.

В итоге в зависимости от результатов анализа, на исходное изображение карточек наносятся соответствующие маркеры, в координаты центров найденных контуров с небольшим смещением, чтобы разметка находилась в пределах границ карточек.



Рис. 11: Пример результата работы программы.

Описание программной реализации

Программа написана на языке Python с использованием библиотек `pintpy`, `matplotlib`, `PIL`, `opencv`, `scikit-image` и `scipy`. Для запуска программы необходимо открыть соответствующий `.ipynb` файл, например, в таких программах, как Jupyter Lab или Jupyter Notebook, или же через браузер в соответствующих облачных сервисах, например, Google Colab и т. д.. Входные данные для загрузки можно расположить в рабочей директории программы или в любом другом месте, но указав при этом в соответствующей ячейке кода нужный путь. При использовании Google Colab можно использовать изображения, загруженные в Google Drive. Результаты своей работы программа сохраняет в текущую рабочую директорию. Путь сохранения также можно изменить в соответствующей ячейке в коде. Также при желании можно сохранять промежуточные результаты (в соответствующих демонстрационных ячейках).

Программа разбита на функции перевода изображения в одноканальное, построение маски сегментации карточек, детекции фигур, нахождения контуров и вершин полигонов, анализа фигур и построения выходного изображения. Есть возможность задания таких параметров как способ построения одноканального изображения, размер окна медианного фильтра, радиус диска эрозии для сегментации карточек, степень размытия в детекторе Кэнни в сегментации карточек и фигур, примитивы замыкания и эрозии для фигур и их размеры, максимальный размер удалаемых малых объектов, точность приближения при построении полигона по контуру, пороговое медианное значение угла в аппроксимирующем полигоне для его классификации. Есть и возможность изменения других параметров непосредственно в соответствующих ячейках.

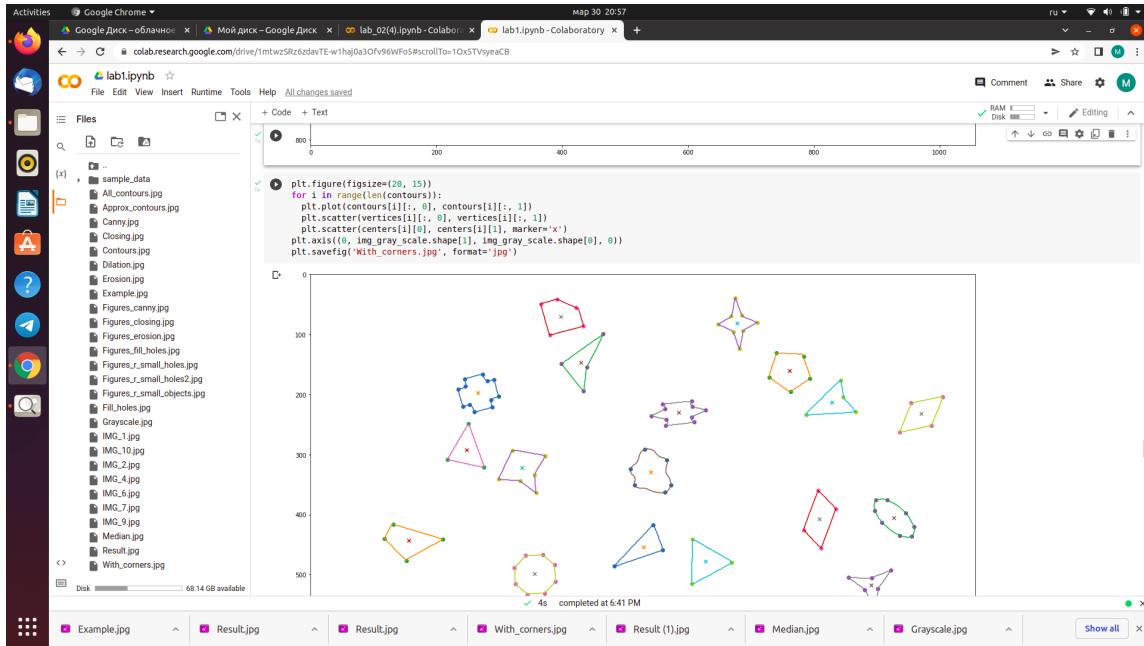


Рис. 12: Работа программы, запущенной в Google Colab

Эксперименты

Рассмотрим результаты работы на нескольких входных изображениях. Программа безошибочно находит число карточек (фигур на них соответственно), точно распознает невыпуклые многоугольники и их число сторон. В случае выпуклых многоугольников проблемы возникают лишь в случае правильных или похожих на них многоугольников, обычно более чем с 5 сторонами. Это происходит в силу используемой эвристики определения гладкости фигуры, т. к. похожими фигурами приближаются окружности и т. д., и у них могут быть довольно большие углы. Из-за нее часто шестиугольники, восьмиугольники и т. п. распознаются как гладкие фигуры, тогда как грушевидный гладкий контур или «облако» могут распознаться как выпуклый многоугольник в силу наличия участков с относительно большой кривизной. Таким образом, подобрать пороговое медианное значение угла для классификации сложно, однако можно варьировать его для получения лучшего из возможных результатов. Что касается остальных фигур, то с их классификацией проблем не возникало, разве что с невыпуклой 10-ти угольной фигурой, чем-то похожей на лезвие. Только на одном из изображений на ней было посчитано на 1 вершину меньше, чем на самом деле. Однако данная ошибка решилась изменением размера примитива эрозии, при этом классификация остальных фигур осталась верной. Вообще на 5 тестовых изображениях (без сложного фона) на одном наборе

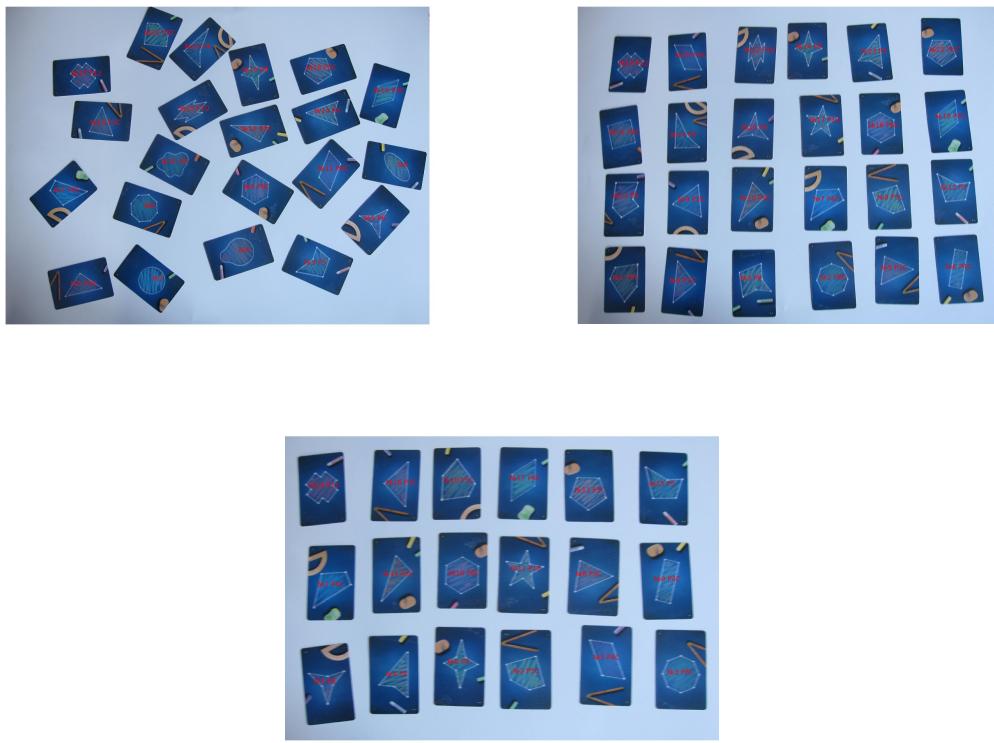


Рис. 13: Примеры результатов работы программы на разных входных изображениях.

параметров удалось провести точную классификацию многоугольников, только с 1 ошибкой на «лэзвии». Что касается гладких фигур, то при их классификации обычно совершается больше или равно 1 ошибки, обычно в паре с ошибкой на 6-ти или 8-миугольнике (он определяется как гладкая кривая). При варьировании порогового значения угла данная проблема полностью не решается. Можно предположить, что в зависимости от входного изображения настройкой параметров можно достигнуть близкого к абсолютно точному результата, в случае только многоугольников реально получить точную классификацию. Можно также пробовать различные примитивы для замыканий и эрозий для получения границ фигур лучшего качества, или брать разную точность аппроксимации полигоном. Также при экспериментировании с различными фильтрами для выделения границ все методы давали результат значительно хуже, чем детектор Кэнни. Можно также варьировать его параметр размытия для улучшения результата. Были опробованы и разные способы получения одноканального изображения. Выделение красного канала на данных изображениях дало наилучшие результаты, наиболее четкие границы. Также рассматривалось использование различных преобразований гистограмм, но весомого влияния это не оказалось.

Выводы

В результате данной лабораторной работы была разработана программа для сегментации карточек игрового набора «Геометрика», их подсчета и определения фигур на них. Разработанный алгоритм в целом способен работать и с другими подобными геометрическими объектами. Метод способен давать идеально точный результат для многоугольников, как выпуклых, так

и невыпуклых. Задача по детекции гладких фигур не была решена идеально точно, возникает проблема подбора порогового значения угла в силу особенностей используемой эвристики. Варьируя его, можно добиться приемлемого результата, однако нужно будет решать проблему «размена» ошибок на многоугольниках или на гладких контурах. Таким образом, требуется тщательный подбор гиперпараметров, либо же использование или разработка более точных и мощных методов определения гладкости контура. Разработанная программа справляется с задачей в случае изолированных и перекрывающихся карточек одинаково хорошо, однако приемлемые результаты были получены лишь для простого фона. Также стоит отметить, что метод устойчив к наличию сторонних объектов по краям карточек, и присутствует гибкость в выборе значений параметров. В итоге можно сказать, что метод требует усовершенствований и более тщательной отладки для получения абсолютно точного результата и работы со сложным фоном. Однако текущая реализация очень хорошо справляется с задачей, допуская не более 2-3 ошибок классификации, что говорит о состоятельности алгоритма.