

Московский Государственный Университет

Сегментация и обработка изображений

Выполнил: Курцев Д.В.
Группа: 317

Факультет Вычислительной математики и кибернетики
Кафедра Математических методов прогнозирования

Март 2022

1 Постановка задачи

На вход программе подаются изображения карточек игрового набора Геометрика:



Необходимо вывести количество карточек (они могут быть и изолированы друг от друга, и накладываться). По запросу пользователя производится распознавание фигур изображённых на рисунке. Для них определяется количество углов и проверка на выпуклость многоугольника. Также программа работает с изображениями с монотонным и пёстрым фоном.

2 Ход работы и программная реализация

В качестве программного инструмента будем использовать библиотеку *OpenCV*.

Предобработка

Так как на изображениях много лишних предметов (стирательные резинки, транспортиры, циркули и прочее), выделим и удалим их. Для этого считаем картинку и представим её в *HSV* формате. Функцией *inRange* детектируем только синие области и получим маску нашего изображения.



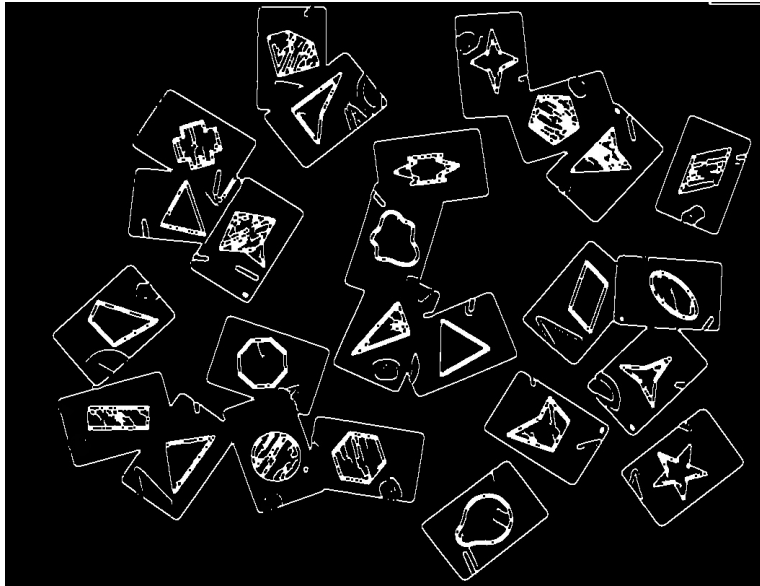
mask

Далее с помощью *Canny* считаем контуры предметов, изображённых на нашей картинке, представленной в чёрно-белом формате и умноженной на маску. Но перед этим применим гауссово размытие, чтобы избавиться от бликов и ненужных шумовых контуров (*GaussianBlur*).



contours

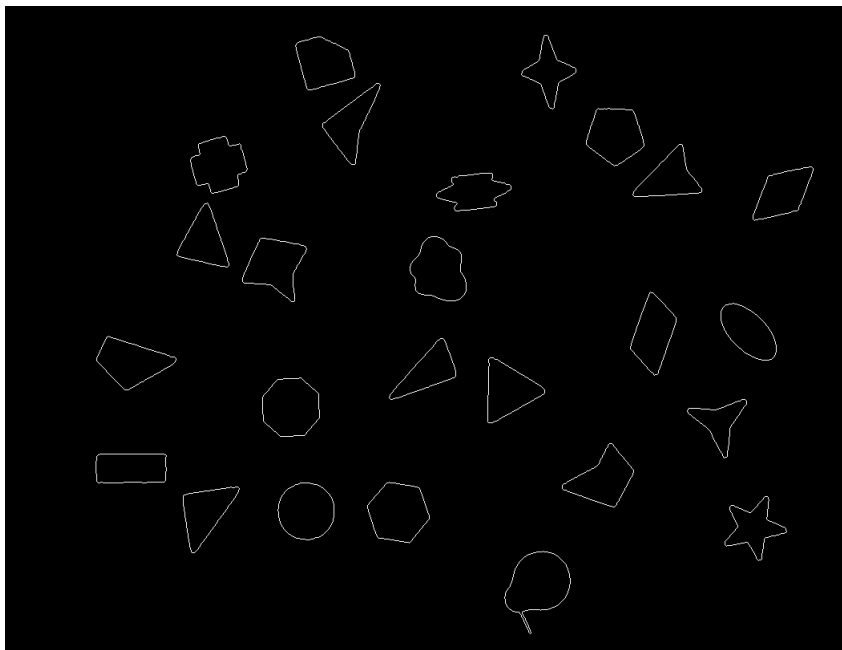
Заметим, что некоторые контуры немного обрываются. Чтобы исправить это и ещё больше подчеркнуть их используем дилатацию и замыкание изображения. Воспользуемся функциями *dilate* и *morphologyEx*.



closed

Beginner

Далее получим контуры фигур с помощью функции *findContours*. Причём мы выберем только внешние с помощью флага *RETR_EXTERNAL*. Мы возьмём только нужные нам контуры (не будем выводить карты и т.д.). Для этого отберём их по площади ограничивающих фигур и периметру. Будем использовать функции *contourArea* и *arcLength* (здесь считается, что размеры карт и фигур одинаковы).

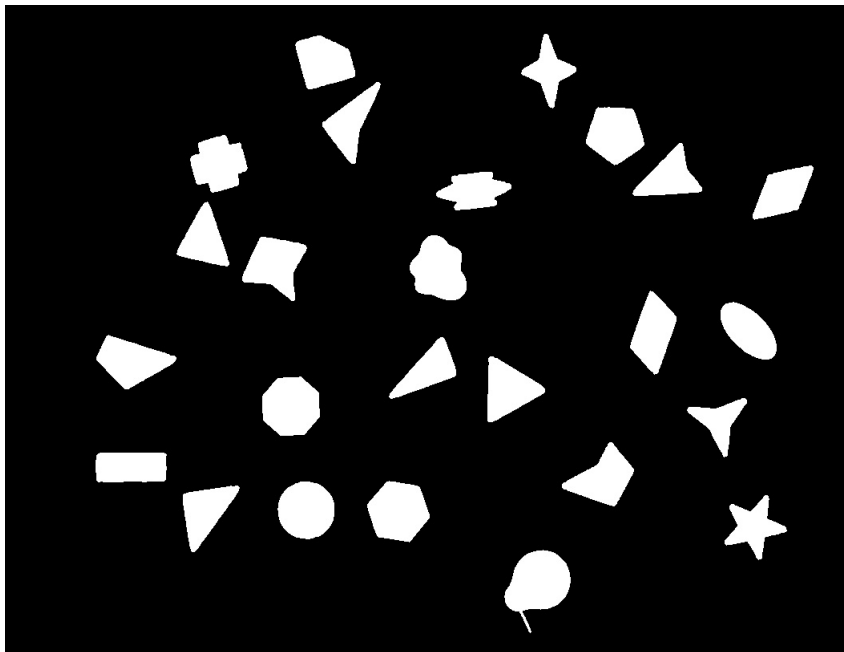


contours_of_figures

На этом этапе уже можно найти число карт на изображении. Это количество изображённых контуров.

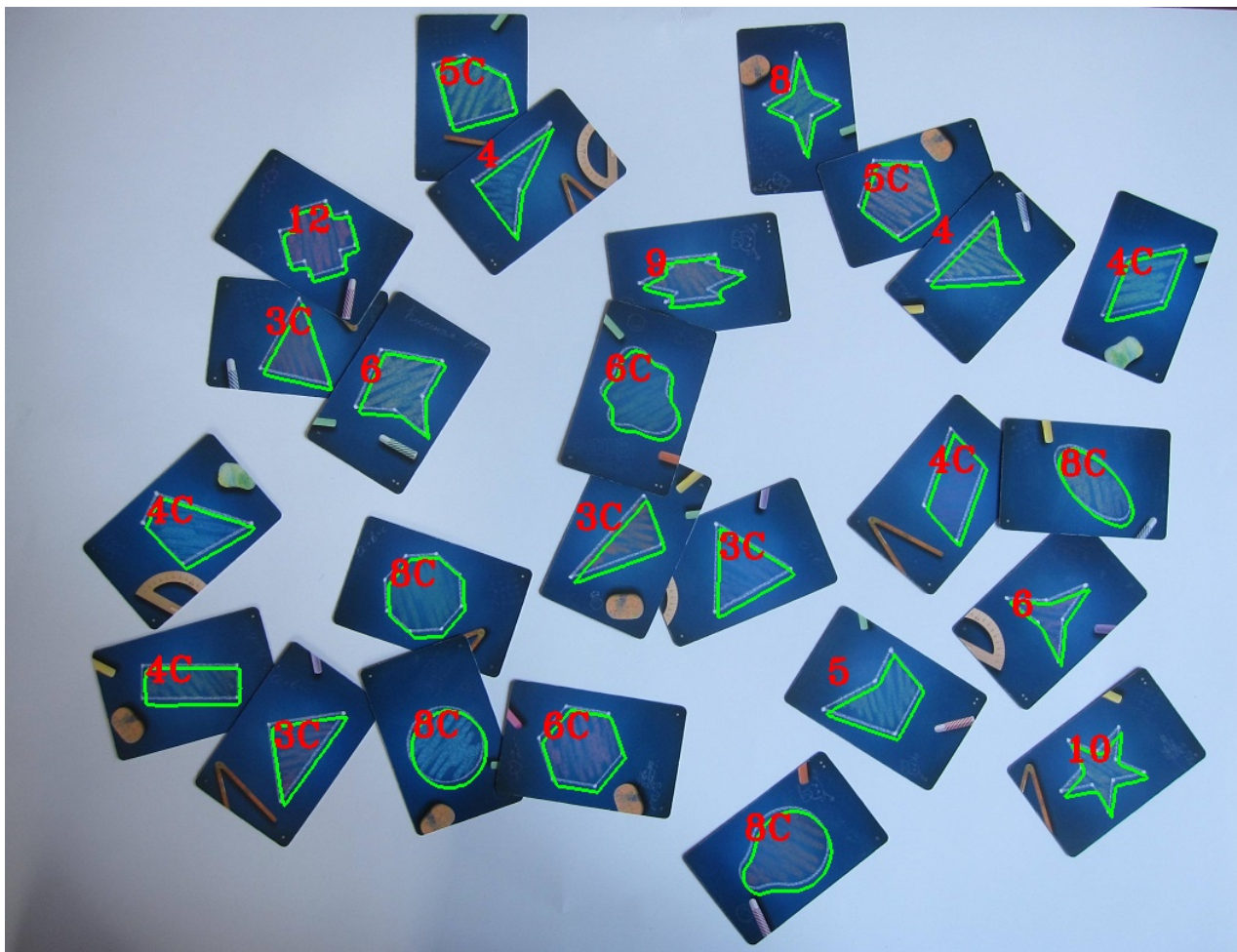
Intermediate

Чтобы дальше правильно обработать фигуры необходимо сгладить их границы, которые получились не очень ровными. Для этого сначала закрасим фигуры с помощью функции *drawContours*, передав флаг *thickness = cv2.FILLED*. Чтобы сделать контуры более гладкими снова воспользуемся методом дилатации. А далее получим более чёткие углы благодаря эрозии и методу *erode*:



eroded

Снова получим границы наших образов методом *findContours*. С помощью функций *approxPolyDP* и *isContourConvex* узнаем сколько углов у фигур и определим являются ли они выпуклыми. И нанесём ответы и контуры на итоговую картинку:



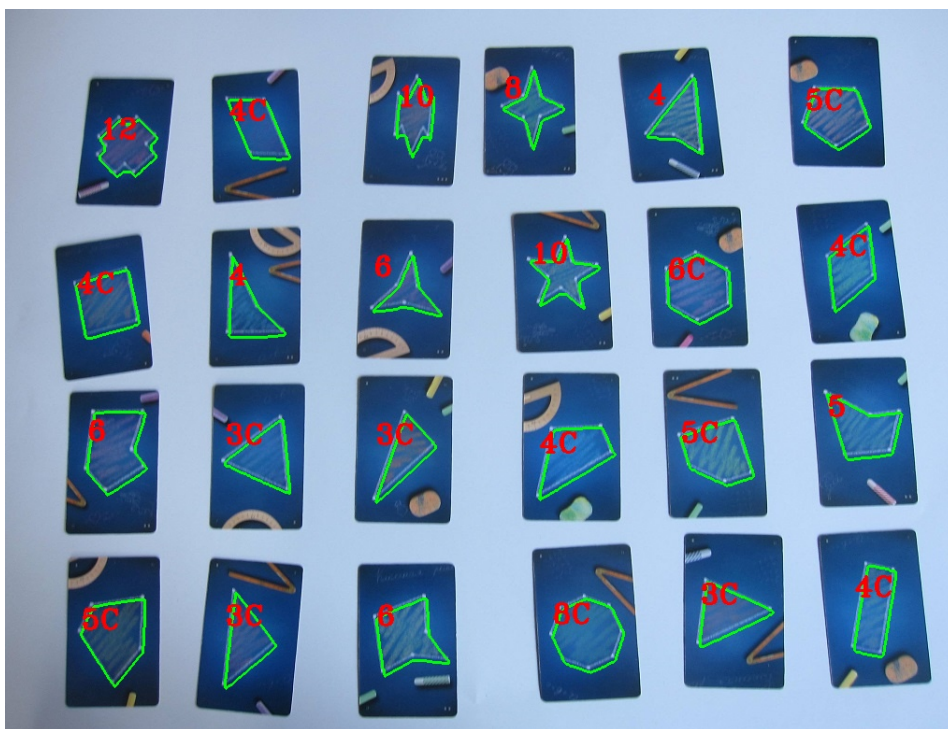
result

Заметим, что программа практически всё верно распознала.

Для удобства пользователя предыдущие пошаговые действия были обернуты в функции *num_cards* и *find_figures*, с помощью которых можно получить ответ. Первая на вход принимает путь до картинки и выдаёт количество карт, изображённых на ней. *find_figures* так же принимает путь и ещё один параметр *draw_cont*. Если он равен *True*, то на изображение помимо ответа будут нанесены контуры фигур. Эта функция возвращает исходную картинку с изображённым на ней результатом (пример "result").

3 Эксперименты

Продemonстрируем работу нашей программы на нескольких изображениях:



IMG_6



IMG_9



IMG_10

Заметим, что наша программа всегда правильно определяет количество карт на изображении и верно детектирует границы фигур, нарисованных на них. Так же она всегда верно делает проверку на выпуклость многоугольника. Но немного ошибается в определении углов(максимум 1 раз на картинке). К примеру, третья фигура в верхнем ряду, изображённая на рисунке *IMG_6*. Там было верно определено количество углов - 10, как и на *IMG_10*. Но на рисунке *result* нет. А на *IMG_9* вообще неверно нарисован контур. Вероятно такие ошибки связаны с чёткостью границ фигур на исходном изображении и качеством фотографии. Например, на *IMG_9* нижняя область сильно засветлена, поэтому невозможно верно определить фигуру.

Так же программа, например, ошиблась при определении прямоугольника на *IMG_10*, сказав, что у него 6 углов. Хотя, как мы видим, на других изображениях он был верно распознан. Так же можно заметить, что на *IMG_10* у него правильно обведена гра-

ница. Вероятно, ошибка связана с недостаточной аппроксимацией контура. Но как было эмпирически проверено, если ещё больше увеличить эрозию, то программа начинает ошибаться на других фигурах.

4 Выводы

Для полной корректности работы программы необходимо подавать изображения в хорошем качестве, без бликов. Иначе возможны небольшие погрешности в работе. Но в большинстве случаев удаётся полностью точно определить границы и тип фигур.

P.s. Проверка на гладкость фигуры не была проведена, так как контуры окружностей и восьмиугольника очень близки друг к другу, поэтому никаким из придуманных способов не удалось их отличить...