

Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра Автоматизации Систем Вычислительных Комплексов

Отчёт

по 1 практическому заданию курса
«Обработка и распознавание изображений»

студента 320 группы

Кондранина Дениса Сергеевича

Москва, 2017

Постановка задачи

Разработать и реализовать программу для работы с изображениями фишек игрового набора Тримино. Программа должна обеспечить:

- Ввод и отображение на экране изображений в формате BMP
- Сегментацию изображений на основе точечных и пространственных преобразований
- Поиск фишек на картинках
- Классификацию фишек на картинках

В задание входят две задачи на изображениях разной сложности:

1. Определить положение фишек на изображении
2. Определить маркировку фишек на изображении

Сложность изображений соответствует трём классам: Beginner, Intermediate, Expert.

- Класс Beginner: Фишки расположены на светлом фоне, картинки типа Pict_1_1 и Pict_1_2
- Класс Intermediate: Фишки расположены на синем фоне с неоднородным освещением, картинки типа Pict_2_1, Pict_2_2 и Pict_2_3
- Класс Expert: Фишки расположены на пестром фоне с неоднородным освещением, картинки типа Pict_3_1, Pict_3_2, Pict_4_1, Pict_4_2

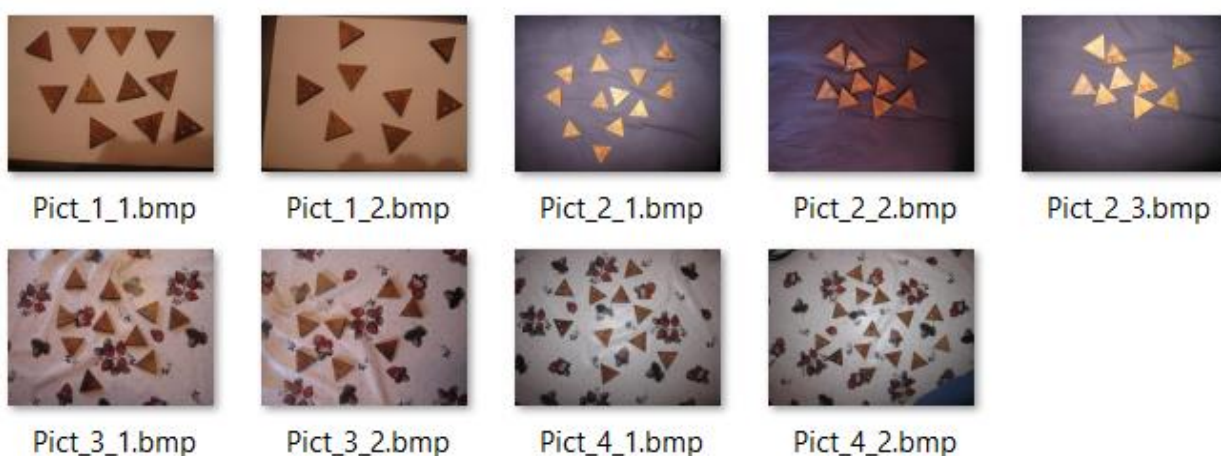


Рис. 1. Предлагаемые изображения для отладки программы

Предлагаемый способ решения задачи



Данную задачу разобьём на две подзадачи

- Распознавание координат триминошек на изображении
- Определение класса триминошки

Распознавание треугольников



Рис. 3. Исходное изображение

Изначально применим медианный фильтр к исходному изображению. Далее рассмотрим H-компоненту изображения пространства HSV. Применим к ней медианный фильтр (размер ядра = 5). Опишем метод распознавания треугольников (соответственно, и триминошек):

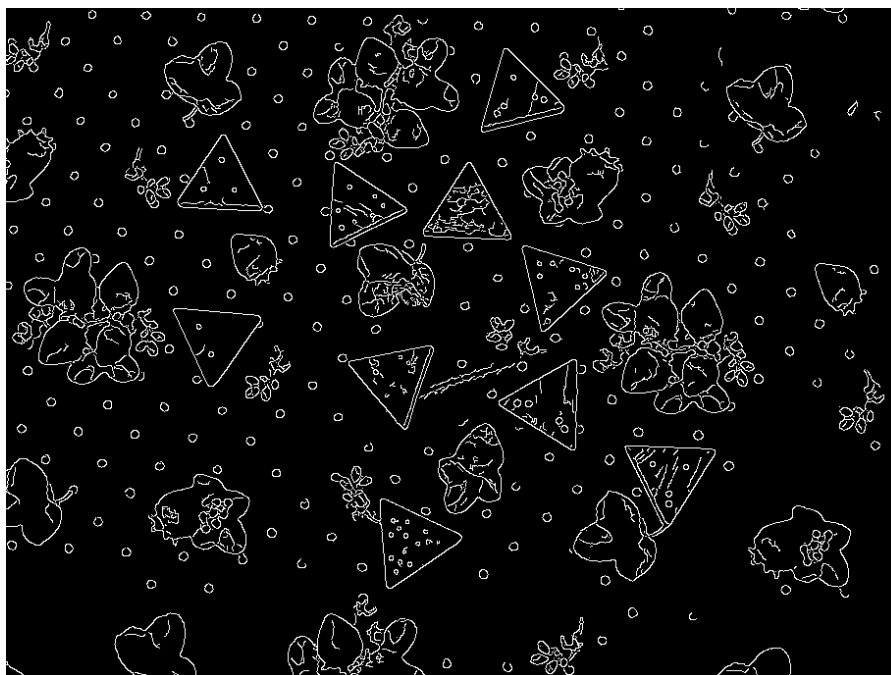


Рис. 4. Границы изображения.

Так как треугольники могут соприкасаться, распознавание с помощью выделения связных компонент или сегментации затруднено. Хорошим способом в таком случае будет распознавание треугольников на основе трёх прямых линий, которые ещё называются «сторонами» треугольника.

Для распознавания прямых линий воспользуемся преобразованием Хафа к границам изображения. Рассмотрим пересекающиеся блоки размера 150×150 , блоки берутся с шагом 10 (по условию задачи известен средний размер стороны треугольника).

Далее в каждом из блоков применим преобразование Хафа для распознавания прямых линий. Из всех найденных прямых линий выберем три, которые образуют треугольник со сторонами, которые отличаются друг от друга не более, чем на 20 пикселей (треугольник должен быть как можно более равнобедренным).

Получившийся список треугольников фильтруется (из всех треугольников, чьи центры близки, выбирается наиболее вероятный, вычисляется средняя Н-компонента треугольника и «доверие» к треугольнику — минимальное (по стороне) отношение морфологически расширенных Кэнни-границ стороны треугольника к длине стороны треугольника). Треугольники со средним значением цвета больше 50 или с доверием меньше 30% не рассматриваются.

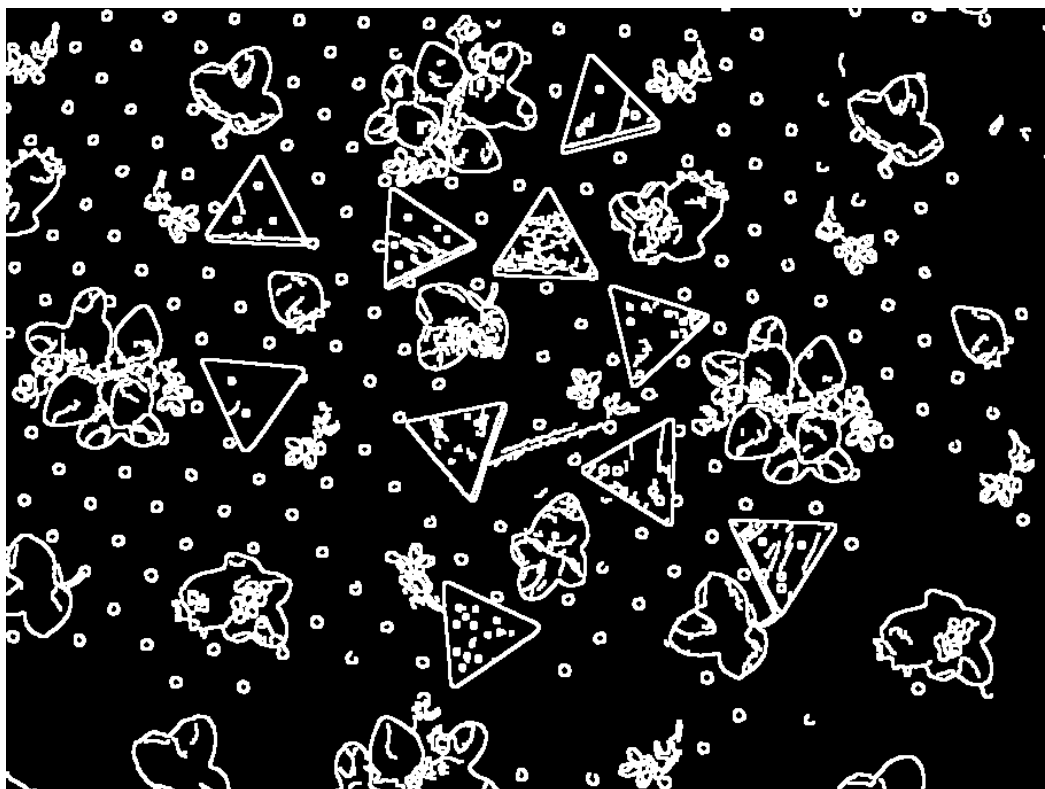


Рис. 5. Результат применения морфологической операции открытия к границам на изображении (см Рис. 4)

Итоговое множество треугольников далее рассматривается как искомые триминошки.



Рис. 6. Результат работы на первом этапе алгоритма

Определение класса триминошки

Для определения класса триминошки будем использовать цветовые и геометрические признаки. Сначала найдём потенциальные точки на триминошках

- Для распознавания белых точек построим карту как минимум из каждой RGB-компоненты
- Для распознавания жёлтых точек построим карту как разность минимума R,G-компонент и B-компоненты
- Для распознавания синих точек построим карту как разность значения B-компоненты и максимума из (R, G) компоненты
- Для распознавания красных точек построим битовую карту, соответствующие пиксели исходного изображения должны удовлетворять условию:

$$(I_h(x, y) < 5 \text{ or } I_h(x, y) > 170) \text{ and } I_s(x, y) > 160 \text{ and } I_v(x, y) > 80,$$

где I_h, I_v, I_s — HSV — компоненты исходного изображения

- Для распознавания зелёных точек построим карту как инвертированная карта минимумов из каждой RGB-компоненты (то есть, по сути, распознаём чёрные точки)

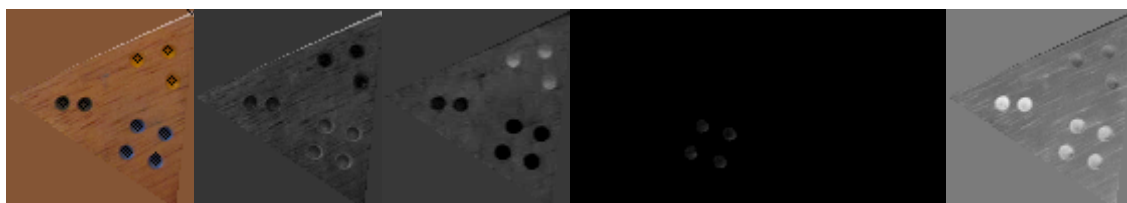


Рис. 7. Слева направо: исходное изображение с обозначенными точками, карта белых точек, карта жёлтых точек, карта синих точек, карта красных точек, карта зелёных точек

С помощью бинаризации по порогу, а также box-фильтра для каждой карты определим список потенциальных точек (как локальные максимумы)

Каждую точку соотнесём ближайшей вершине. Если точка находится совсем близко к вершине, то удалим её.

С помощью следующих свойств найденных точек отклассифицируем триминошки:

- Белые, жёлтые и синие точки распознаются безошибочно. Поэтому, если найдена подобная точка у угла, то метрики расстояния до других классов штрафуют
- Красные точки находятся почти так же, как и предыдущие, но чуть-чуть хуже. Поэтому и штрафовать их будем чуть-чуть поменьше

К сожалению, подобный метод не позволяет грамотно разделить зелёные точки от отсутствия точек.

Описание программной реализации

Предложенный метод был реализован с помощью языка программирования Python3, использовались библиотеки numpy, opencv, os, math, в процессе разработки использовалась ОС Linux Mint Sarah.

Изображения, на которых будет происходить тестирование, можно положить в папку input, тогда в папке output после выполнения программы появится текстовый файл с результатом выполнения программы.

Запускать код следует как “python3 start.py”.

Код доступен по ссылке: <https://github.com/denyhoof/cmc6.images1>

Точность работы программы

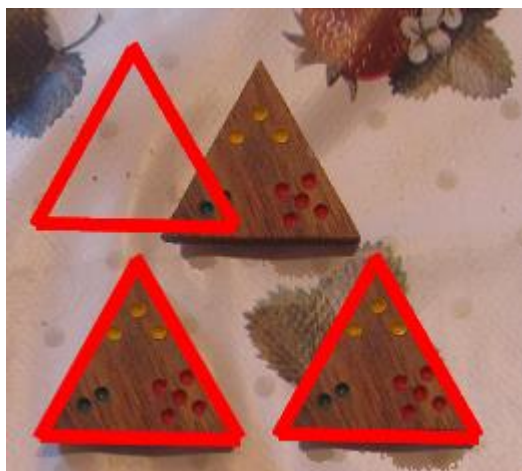


Рис. 8. Ожидаемая точность после первого запуска программы.

Программа тестировалась **только** на выборке Expert.

Точность выбора треугольников — 100%

Точность определения класса вершины треугольника — 92.8%

При этом все ошибки определения класса произошли как раз из-за неверной классификации между отсутствием точек и двумя зелёными точками.



Рис. 9. Пример неверной работы программы

Заключение



Рис. 10. «Дорогая принцесса Селестия, сегодня я усвоил урок, что отчёт следует писать не за час до дедлайна»

В данной работе был предложен алгоритм распознавания и классификации игровых фишек набора Тримино. Итоговая точность на выборке "Expert":

- Точность выбора треугольников — 100%
- Точность определения класса вершины треугольника — 92.8%