

Лабораторная работа №2. Изучение и освоение методов классификации формы изображений.

Петренко Дарья, 317 группа

19 мая 2019

Содержание

Постановка задачи	2
Описание данных	2
Описание метода решения	2
Описание программной реализации	3
Эксперименты	4
файл 2.jpg	4
файл 3.jpg	4
файл 4.jpg	5
файл 5.jpg	6
Выводы	6

Постановка задачи

Для решения выбрано задание уровня Intermediate.

Задание предполагает работу с изображениями графов, построенных из деталей магнитной игры-головоломки, на однотонном фоне. Всего задано 4 различных структуры графа (эталонные образцы предоставлены). Каждый из изображенных графов изоморфен одному из эталонов; таким образом, все графы делятся на 4 класса.

Необходимо по входному изображению определить, к какому классу относится соответствующий граф.

Описание данных

Часть предоставленных данных необходима только для решения задания уровня Expert, в данной работе эти файлы использованы не были.

Для обучения и отладки программы были предоставлены файлы в формате JPEG 1024x768 с разрешением 72 dpi, а также текстовый файл, содержащий таблицу соответствия имен файлов номерам классов изображенных графов. Также в этом файле приведена информация о фоне изображений (белый фон - для задач класса Intermediate, пестрый - для задач класса Expert).

Описание метода решения

Было замечено, что номер класса, к которому принадлежит граф, однозначно определяется информацией о количестве замкнутых циклов, а также о числе компонент связности, которые образуются после удаления всех вершин и ребер, принадлежащих этим циклам. А именно: граф класса 1 содержит 4 замкнутых цикла, граф класса 3 - 5 замкнутых циклов. Графы классов 2 и 4 содержат по 2 цикла, но после удаления принадлежащих циклам вершин и ребер в графе класса 2 остается 4 компоненты связности, а в графе класса 4 - 5 компонент. Разработанный алгоритм направлен на получение описанной информации.

Изображение считывается из файла в черно-белом виде и бинаризуется по порогу так, чтобы пиксели фона были нулевыми, а пиксели графа единичными. После такой операции пиксели, соответствующие случайным затемнениям по краям изображения, иногда тоже становятся единичными. Для исправления этой ошибки рамка по краям изображения шириной в 20 пикселей полностью выставляется в ноль.

Для удаления небольших промежутков в получившемся контуре графа применим операцию замыкания. Построим скелет изображения, а затем для корректного нахождения внешних и внутренних контуров изображения увеличим ширину его линий с одного пик-

селя до трех (применим операцию дилатации). Теперь можно применить операцию поиска контуров изображения. Удалим все контуры, периметр которых меньше некоторого небольшого числа, чтобы убрать случайных мусор. В результате получим один внешний контур графа и несколько внутренних, по одному на каждый замкнутый цикл. Если количество контуров равно 5, то сразу можно утверждать о принадлежности графа классу 1. Если оно равно 6, то классу 3. Для случая, когда контуров 3, проводим дальнейший анализ.

Сгенерируем двумерную нулевую матрицу, размер которой совпадает с размером изображения. Будем хранить в ней сумму изображений, полученных в цикле. Пройдясь по списку контуров, для каждого сгенерируем аналогичную нулевую матрицу и нанесем на нее контур, увеличив его ширину до 10 пикселей, так чтобы пиксели фона оставались нулевыми, а пиксели контура имели значение 1. Полученную матрицу прибавляем к матрице суммы. По окончании этой операции пиксели, принадлежащие одновременно нескольким контурам (а это равносильно принадлежности замкнутым циклам графа), имеют значение больше 1. При помощи бинаризации по порогу оставим ненулевыми только такие пиксели.

Применим к полученному изображению операцию дилатации, увеличив толщину линий полученных замкнутых контуров. Затем, используя его в качестве маски, обнулим на матрице скелета пиксели, принадлежащие циклам. Осталось только посчитать количество получившихся связанных компонент, периметр которых больше некоторого небольшого числа, чтобы убрать случайный мусор. Если это число равно 4, то относим граф к классу 2. Если оно равно 5, то к классу 4.

Описание программной реализации

Программный код был написан на языке Python с использованием библиотек OpenCV, skimage, numpy.

Бинаризация изображений осуществлялась при помощи функции `threshold` библиотеки OpenCV. Операции замыкания, дилатации, построения скелета - с использованием функций `binary_closing`, `dilatation`, `skeletonize` библиотеки `skimage.morphology`. Работа с контурами осуществлялась при помощи функций `findContours` и `drawContours` библиотеки OpenCV (поиск контуров и нанесение их на изображение). Связные оболочки были найдены функцией `label` библиотеки `skimage.measure`.

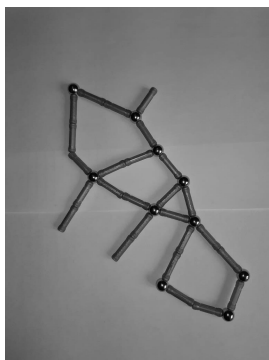
Эксперименты

файл 2.jpg

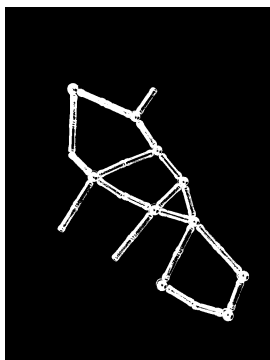
Вывод программы: Граф класса 1.

Ниже представлены изображения, полученные на промежуточных этапах обработки:

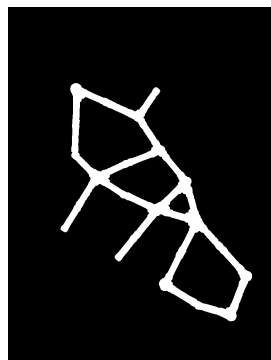
(a) - исходное изображение, (b) - после бинаризации, (c) - после замыкания бинаризованного изображения, (d) - скелет, толщина линий которого увеличена до 3.



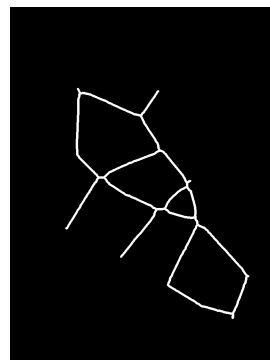
(a)



(b)



(c)



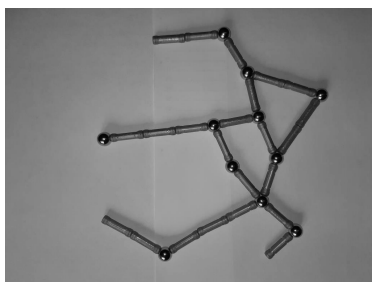
(d)

файл 3.jpg

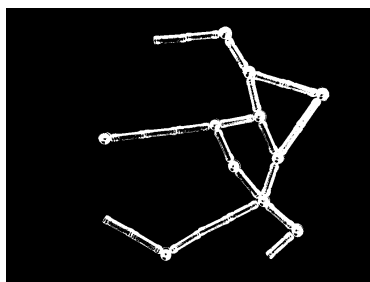
Вывод программы: Граф класса 2.

Ниже представлены изображения, полученные на промежуточных этапах обработки:

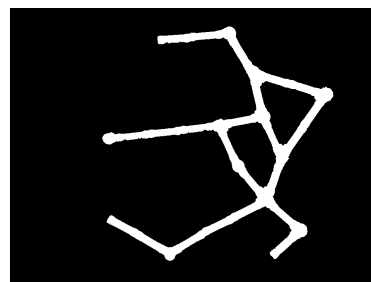
(a), (b), (c), (d) аналогичны описанным в предыдущем эксперименте. (e) - результат суммирования контуров (можно заметить, что части, относящиеся к циклам, более светлые, так как значения соответствующих пикселей больше 1), (f) - после бинаризации суммы контуров по порогу, (g) - после вычитания циклов.



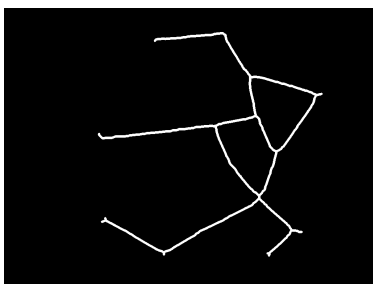
(a)



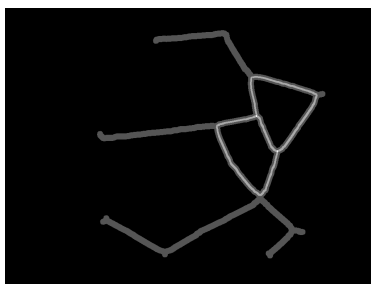
(b)



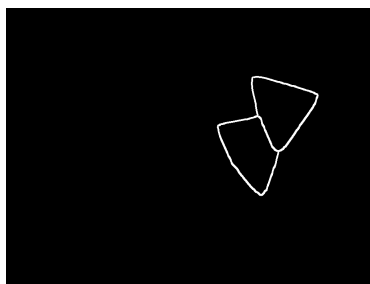
(c)



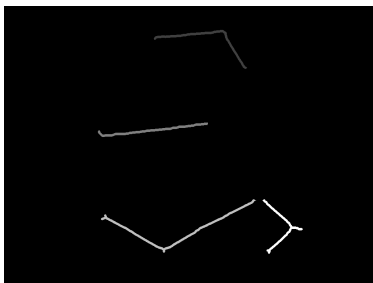
(d)



(e)



(f)

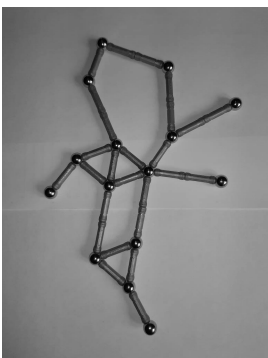


(g)

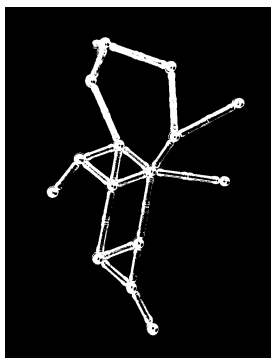
файл 4.jpg

Вывод программы: Граф класса 3.

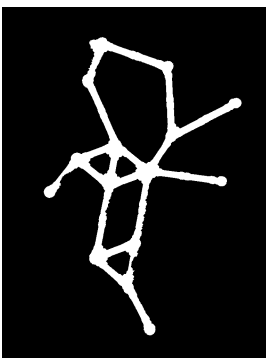
Промежуточные этапы аналогичны описанным в эксперименте 1.



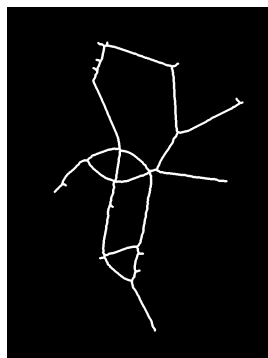
(a)



(b)



(c)

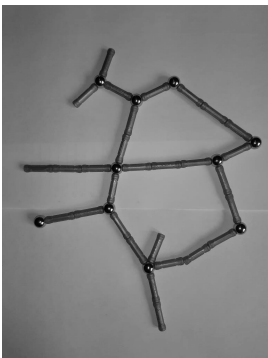


(d)

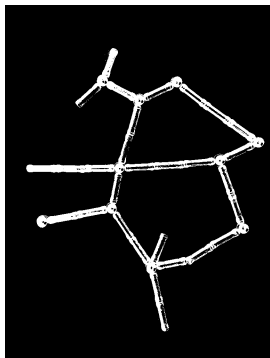
файл 5.jpg

Вывод программы: Граф класса 4.

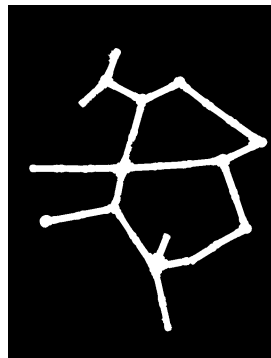
Промежуточные результаты аналогичны описанным в эксперименте 2.



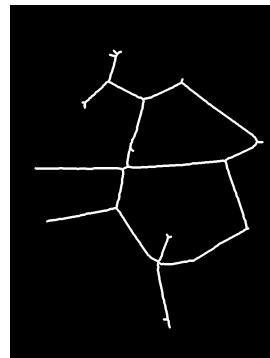
(a)



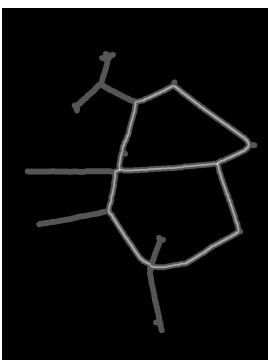
(b)



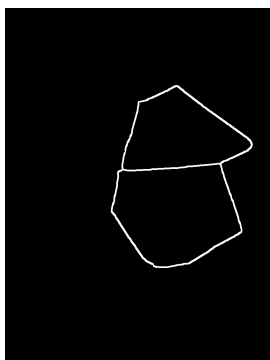
(c)



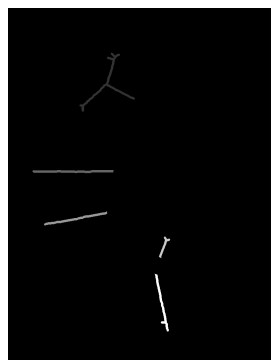
(d)



(e)



(f)



(g)

Выводы

Результаты всех проведенных с использованием обучающей выборки экспериментов положительные, следовательно, разработанный алгоритм работает корректно. Задание выполнено.