

Практическое задание
Изучение и освоение методов классификации формы
изображений.

Кривонос Анна
317 группа

1 мая 2023 г.

Содержание

1	Постановка задачи	3
2	Описание данных	3
3	Метод решения	3
4	Описание программой реализации	4
5	Эксперименты	6
6	Вывод	7

1 Постановка задачи

Необходимо разработать и реализовать программу для классификации изображений моделей графов, построенных из магнитной головоломки. Должен поддерживаться ввод и отображение на экране изображений в формате jpg. Всего задано 4 структуры графа, эталоны которых представлены на рисунках ниже. На остальных изображениях представлены графы, изоморфные четырём эталонным образцам.

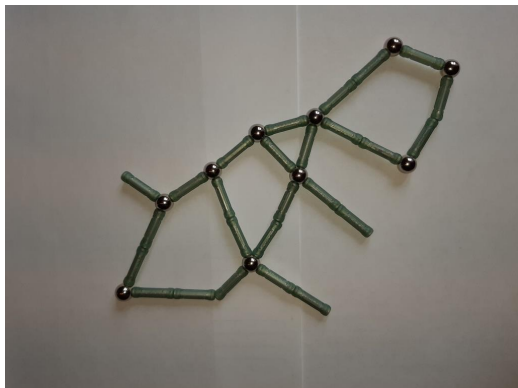


Рис. 1: Тип 1

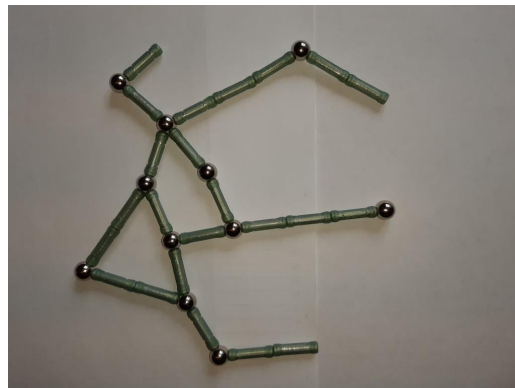


Рис. 2: Тип 2

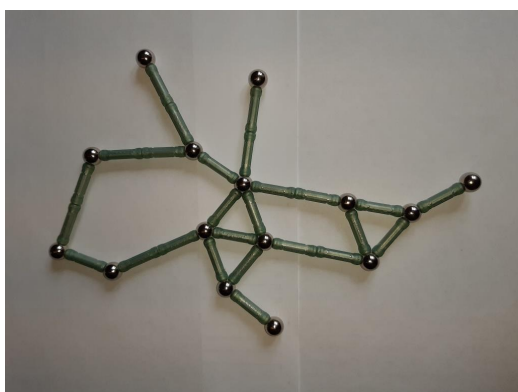


Рис. 3: Тип 3



Рис. 4: Тип 4

2 Описание данных

Данные представляют из себя изображения графа в формате jpg. Вершинами графа являются либо терминальные вершины (степени 1), либо вершины, в которых находится шарик. Все изображения находятся на белом фоне.

3 Метод решения

Заметим, что у каждого типа графа есть следующие отличительные черты:

- Тип 1 и только он содержит три вершины степени 3
- Тип 2 и только он содержит одну вершину степени 4
- Тип 3 и только он содержит одну вершину степени 5
- Тип 4 и только он содержит две вершины степени 4

Таким образом, для классификации достаточно определить количество вершин со степенями ≥ 3 .

Для этого выполним сегментацию изображения, включающую переход к полутоновому изображению, бинаризацию и морфологическую обработку, в результате которой получается бинарное изображение. Затем построим скелет бинарного изображения. Далее по полученному скелету вычисляются координаты вершин графа и координаты концов ребер. После для каждой вершины определяется ее степень и проводится классификация графа указанным выше способом.

4 Описание программой реализации

Программная реализация находится в файле `task_2.ipynb`. Для работы с изображениями используются библиотеки `open_cv`, `plantcv`

Для удобства использования программы написаны функции:

- `read_image(path)` - чтение изображения с именем `path`
- `show_image(image)` - вывод изображения на экран
- `togray(image)` - возвращает бинарное изображение

Пример работы функции бинаризации изображения:

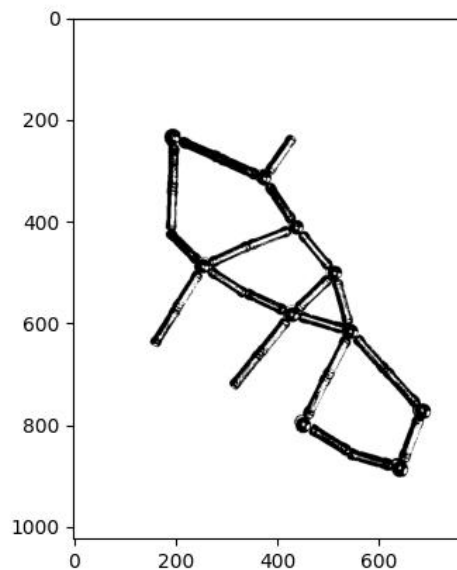


Рис. 5:

Для удаления белых бликов на ребрах применим к изображению несколько раз эрозию и делатацию. В результате получим более четкое бинарное изображение.

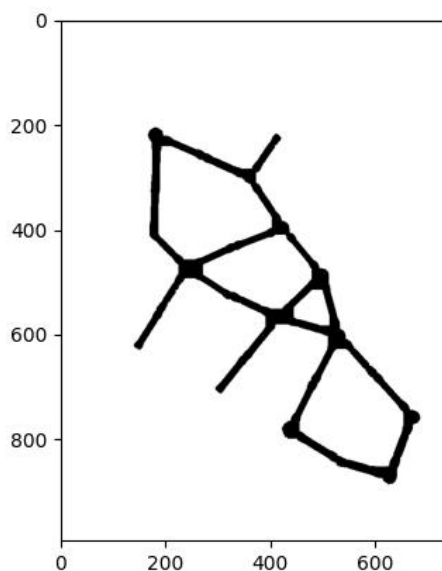


Рис. 6:

Для построение скелета воспользуемся готовой функцией `psv.morphology.skeletonize`. В результате получим следующее изображение:

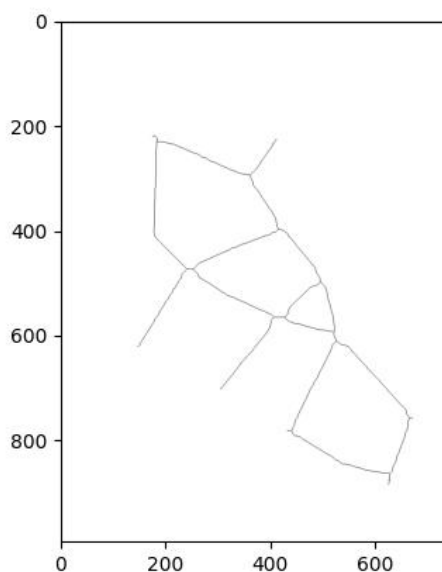


Рис. 7:

Для нахождения координат шариков примени алгоритм делатации к бинарному изображению несколько раз. В итоге у нас останутся черные пиксели только в окрестности шариков. Удалив близкие точки, оставив по одной для каждой вершины, получим искомые координаты. Для нахождения оставшихся вершин(степени 1) воспользуемся функцией `psv.morphology.find_tips`, которая находит координаты терминальных вершин. Объединив два этих множества, получим множество всех вершин графа. Результат работы показан ниже:

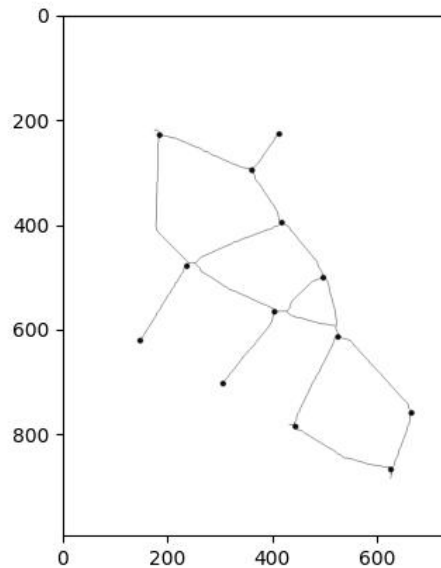


Рис. 8:

Далее с помощью функции `psv.morphology.segment_skeleton` находим контуры ребер, аппроксимируем их и вычисляем координаты концов каждого ребра. Соотносим эти координаты с уже вычисленными координатами вершин (находим для каждого конца ребра ближайшую к нему вершину). Таким образом каждому найденному ребру ставим в соответствие найденные вершины (рис. 8).

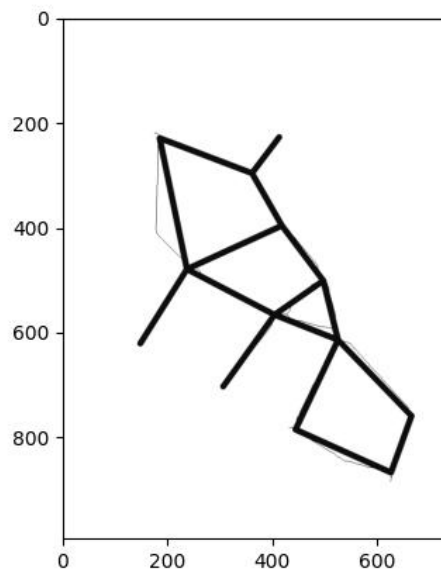


Рис. 9:

Далее для каждой вершины считаем количество ребер, которые к ней прилегают, тем самым вычисляя ее степень.

Итоговая функция `type_graph(image)` применяет к изображению описанный алгоритм и затем классифицирует граф по степеням вершин.

5 Эксперименты

Протестируем наш алгоритм. Для предоставленных изображений получены следующие результаты:

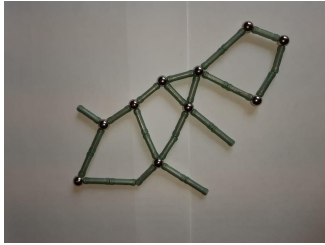


Рис. 10: Тип 1

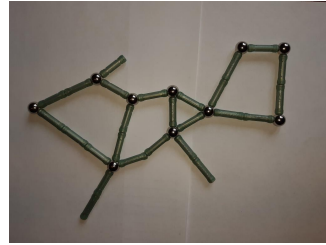


Рис. 11: Тип 1

4

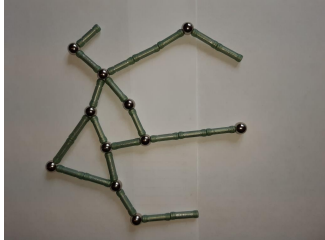


Рис. 12: Тип 2



Рис. 13: Тип 2

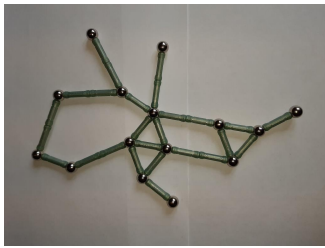


Рис. 14: Тип 3

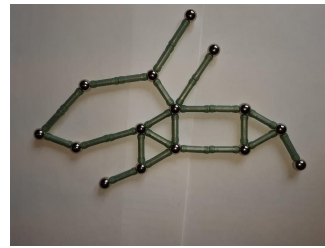


Рис. 15: Тип 3



Рис. 16: Тип 4

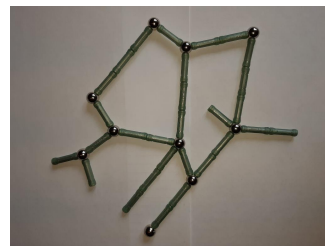


Рис. 17: Тип 4

Видим, что алгоритм правильно классифицирует графы для данного набора картинок.

6 Вывод

Была реализована программа, классифицирующая графы в соответствии с заданными типами. Алгоритм показал корректную работу на имеющихся данных.