

Обработка и распознавание изображений

Лабораторная работа № 2

Изучение и освоение методов классификации формы изображений

Дмитриев Леонид, группа 317

1 мая 2023 г.

# Содержание

Постановка задачи	3
Описание данных	3
Описание метода решения	4
Сегментация графа, получение бинарного изображения . . . . .	5
Intermediate . . . . .	5
Expert . . . . .	6
Построение скелета . . . . .	6
Извлечение из скелета графа вершин и ребер, построение программной модели графа	7
Извлечение из структуры графа вектора признаков . . . . .	8
Классификация графа по имеющимся признакам . . . . .	8
Описание программной реализации	8
Эксперименты	8
Выводы	9

## Постановка задачи

Целью работы является разработка и реализация программы для классификации изображений моделей графов, построенных из магнитной головоломки.

Программа должна обеспечивать:

- Ввод и отображение на экране изображений в формате JPEG
- Сегментацию изображений на основе точечных и пространственных преобразований
- Генерацию признаков описаний структуры графов на изображении
- Классификацию изображений по 4 заданным классам

Задание делится на два вида сложности (Intermediate и Expert). В первом случае фон белый, а во втором пестрый.

## Описание данных

Входными данными являются цветные изображения моделей, из деталей магнитной игры - головоломки в формате  $1024 \times 768$  с разрешением 72 dpi.

Всего задано 4 структуры графов, эталоны которых представлены на 4 имеющихся изображениях.

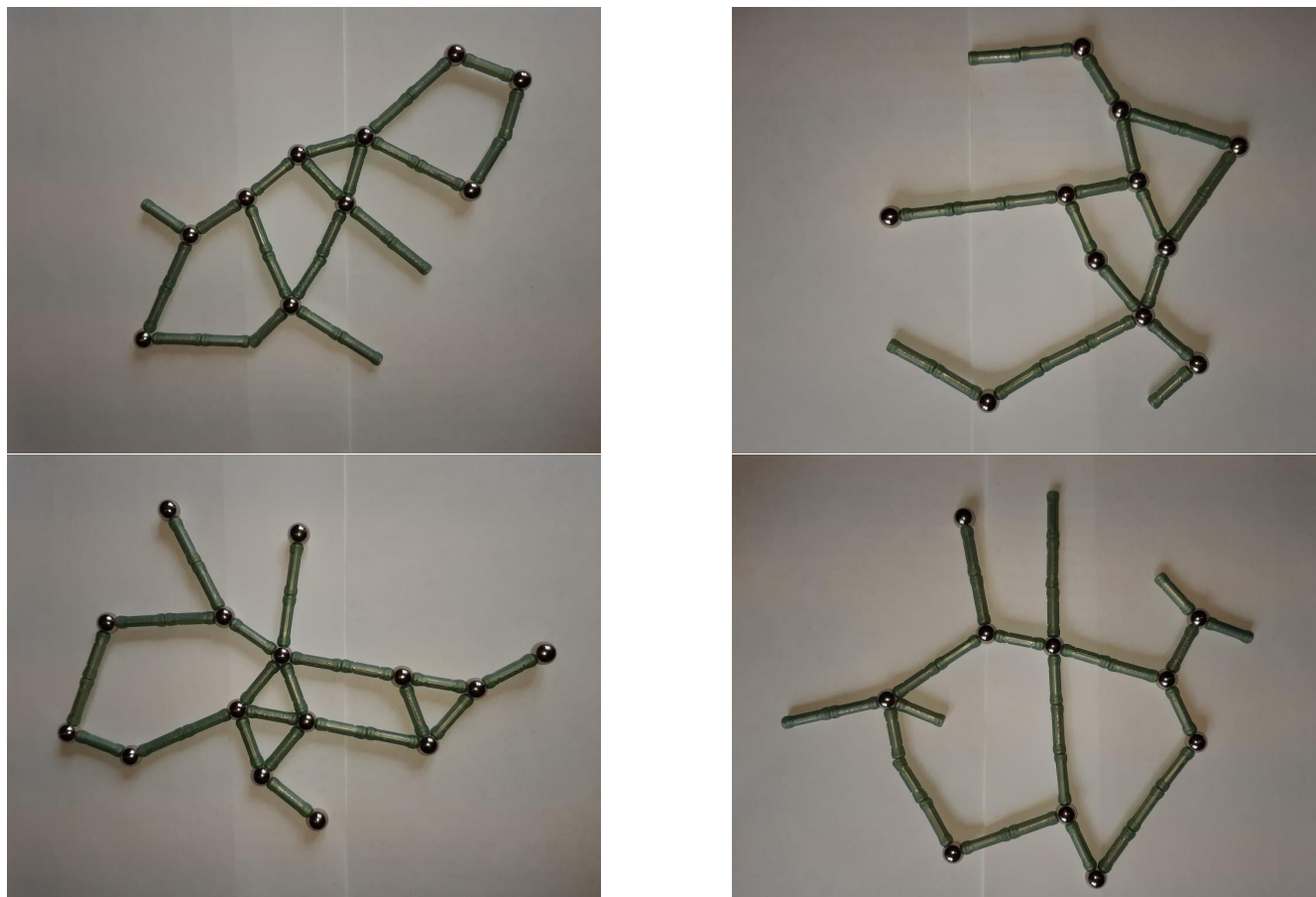


Рис. 1: Изображения эталонов 4 типов графов.

На остальных изображениях представлены графы, изоморфные 4 эталонным образцам. При этом вершинами считаются только узлы степени не равной 2 и длины ребер (веса) не учитываются.

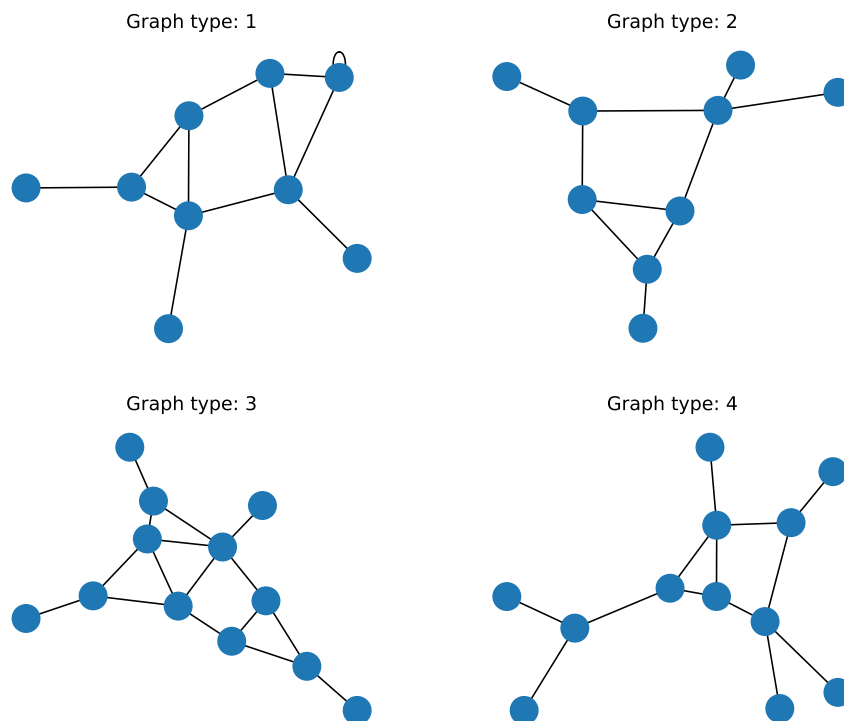


Рис. 2: Изображения типов структур графов.

Для задачи повышенной сложности фон на изображениях разноцветный и с множеством прямых линий.

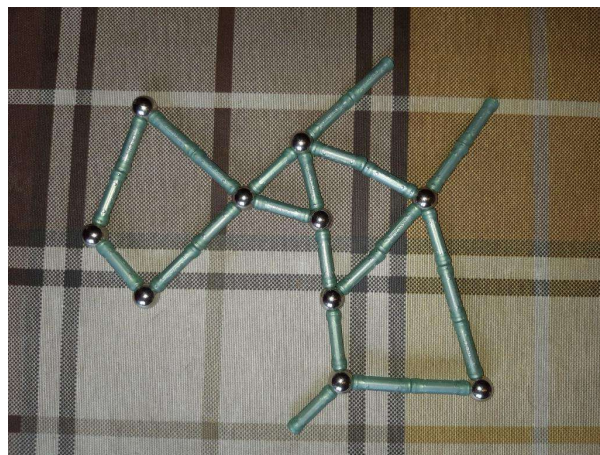


Рис. 3: Примеры изображений повышенной сложности.

## Описание метода решения

Этапы извлечения графа из изображения:

- Сегментация графа, переход к бинарному изображению
- Построение скелета графа
- Извлечение из скелета графа вершин и ребер, построение программной модели графа
- Извлечение из структуры графа вектора признаков
- Классификация графа по имеющимся признакам

### Сегментация графа, получение бинарного изображения

Для обоих уровней сложности реализованы похожие методы решения. Единственным различием является этап бинаризации.

#### Intermediate

Сперва из изображения устраняется шум. Для этого применяется медианная свёртка.



Рис. 4: Исходник (слева), результат медианной свёртки (справа).

Убирать шум необходимо для лучшего обнаружения краев с помощью свертки (функция Canny из библиотеки OpenCV). Также сразу отсекаются по порогам шумовые пиксели, не являющиеся предметами интереса.

Для построения скелета необходимо залить объект, это можно сделать с помощью операции закрытия с большой матрицей свёртки.

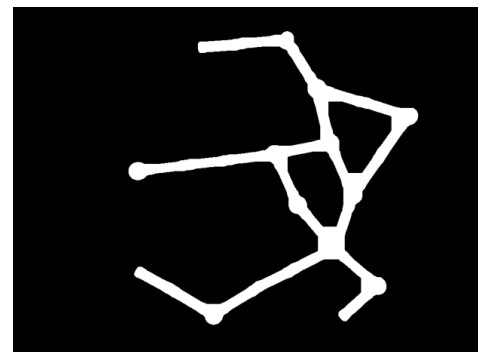
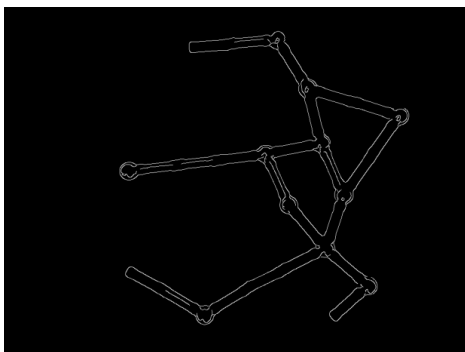


Рис. 5: Результаты обнаружения краёв (слева) и применения закрытия к обнаруженным краям (справа).

## Expert

Сначала изображение переводится в HSV формат, а затем с помощью нескольких масок диапазонов значений сегментируются ребра и вершины по отдельности, после этого маски складываются.

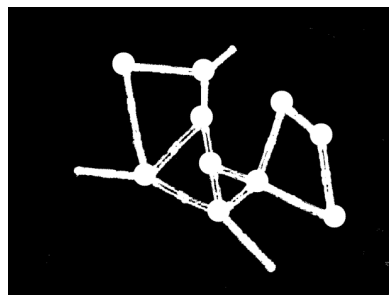
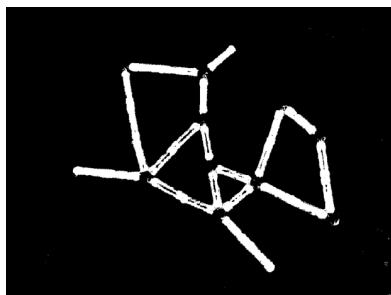


Рис. 6: Результат детектирования ребер и вершин.

Далее удаляется шум с помощью открытия, а затем устраняются внутренние полости с помощью закрытия.

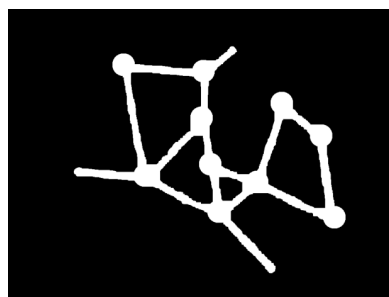
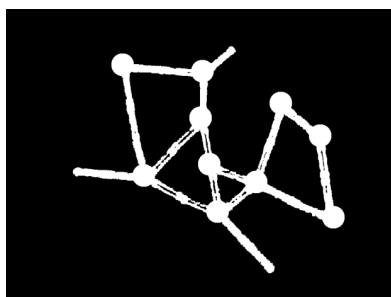


Рис. 7: Результаты удаления шума и удаления полостей.

## Построение скелета

Скелет изображения строится по алгоритму Розенфельда (функция `skeletonize` из библиотеки `skimage`).

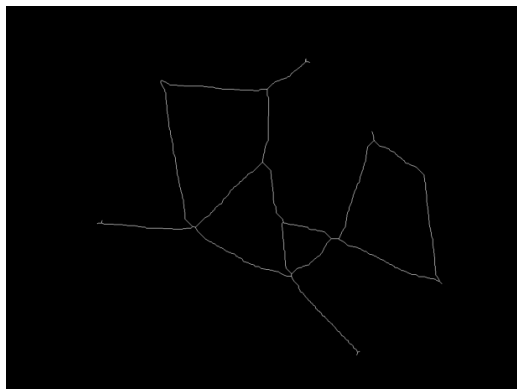


Рис. 8: Пример скелета графа изображения.

## Извлечение из скелета графа вершин и ребер, построение программной модели графа

Была реализована программа по извлечению графа из изображения скелета. Так как функция `skeletonize` использует специфичный тип соседства пикселей было введено специальное правило определения связи между пикселями.

Сильными связями считаются связи по горизонтали и вертикали, если такая связь есть то она учитывается всегда. Слабыми связями считаются связи по диагоналям, такая связь учитывается только если нет обеих прилегающих к этому месту сильных связей.

Реализованный алгоритм возвращает все найденные компоненты связности на изображении в виде графов (множеств из вершин и ребер).

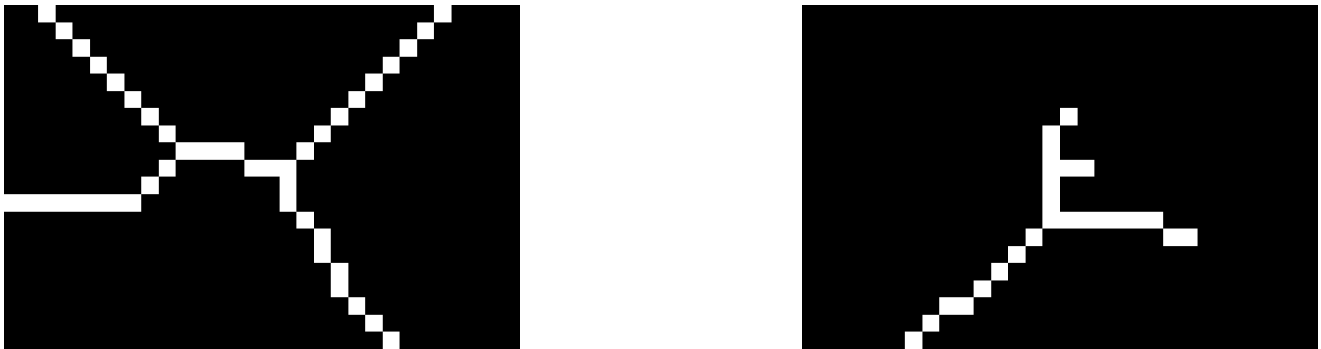


Рис. 9: Пример типичной вершины со степенью 4(слева), пример шумовых ветвей(справа).

Но в итоге построенный граф не является структурно в точности таким же, как на исходном изображении. Это вызвано тем, что при получении скелета происходит расщепление вершин большой степени и появлении мелких шумовых ветвей (показано на рисунке выше). Данная проблема решается поочередным применением операции удаления короткого терминального ребра и операции склеивания вершин связанных коротким ребром. Операции продолжаются до тех пор, пока существуют короткие ребра (короткий - значит длина меньше порога).

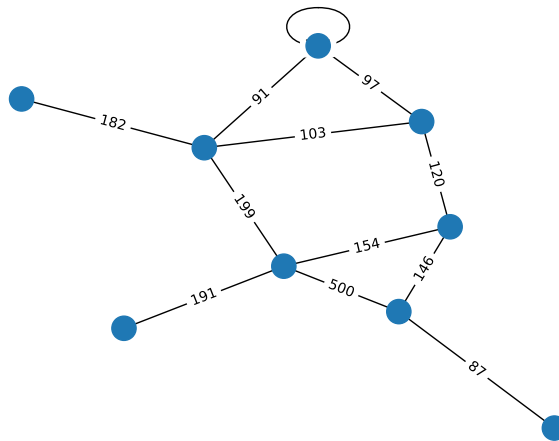


Рис. 10: Пример программной модели.

## Извлечение из структуры графа вектора признаков

Из графа извлекаются следующие признаки: число ребер, число вершин, число терминальных вершин со степенью 1, число вершин со степенью 3, число вершин со степенью 4, число вершин со степенью 5, число циклов в базисе, число петель, число внутренних ребер, кол-во циклов содержащих три вершины.

Для графа разобранного выше (в expert бинаризации) вектор признаков равен:

12, 9, 3, 3, 3, 0, 4, 1, 9, 2

- Для класса 1: [12 9 3 3 3 0 4 1 9 2]
- Для класса 2: [10 9 4 4 1 0 2 0 6 1]
- Для класса 3: [16 12 4 5 2 1 5 0 12 4]
- Для класса 4: [13 12 6 4 2 0 2 0 7 1]

## Классификация графа по имеющимся признакам

Сначала производится попытка проверить изоморфизм полученного графа со всеми заранее найденными эталонными графами, если при построении графа не произошло ошибок, то будет найден верный класс.

Если класс не был найден, то произошла и дальше действуем из предположения, что вектор признаков полученный ранее должен быть ближе всего к вектору признаков верного класса. Таким образом выбираем наиболее близкий класс.

Выбор класса по близости производится с помощью логистической регрессии обученной на 4 эталонных графах, где векторы признаков нормализуются. Этот способ выбран так как линейная модель нативно оценивает важность признаков.

## Описание программной реализации

Программная реализация оформлена в виде юпитер-ноутбука `program.ipynb`, в котором объявляются переменные `FILENAME` и `TASK_TYPE`.

`FILENAME` - необходимо присвоить ей значение имени файла с входным изображением.

Для выполнения решения для задачи Intermediate:

```
TASK_TYPE = "inter"
```

Для выполнения решения для задачи Expert:

```
TASK_TYPE = "expert"
```

После присвоения соответствующих значений необходимо выполнить все строчки кода ноутбука с первой до последней, в итоге в предпоследней ячейке будет напечатан класс входного графа.

## Эксперименты

При проверке на всех тренировочных изображениях алгоритм корректно определял тип входного графа.



## Выводы

Была разработана и реализована программа, определяющая по входному изображению тип графа расположенного на нем среди заданных 4 классов.