

# Лабораторная работа №2

ММП ВМК МГУ

Обработка и распознавание изображений

Егор Черемискин, 317

# Содержание

Постановка задачи	2
Описание данных	2
Метод решения и программная реализация	3
Эксперименты	8
Выводы	8

## Постановка задачи

В данной лабораторной работе требовалось обработать изображения моделей графов, построенных из магнитной головоломки. По входному изображению требовалось построить признаковое описание топологической структуры графа, в нашем случае, вектор чисел, в котором хранится число вершин графа с определенной степенью. По этому признаковому описанию требовалось классифицировать изображение на четыре заранее заданных класса.

## Описание данных

Для тестирования написанной программы предлагалось по 2 изображения на каждый класс, то есть всего 8 изображений. На изображении находится граф из магнитной головоломки на сером фоне.

Ниже изображены все доступные изображения.

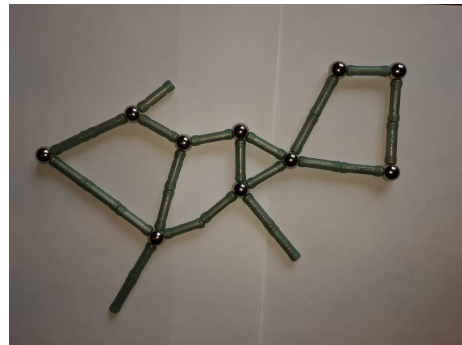
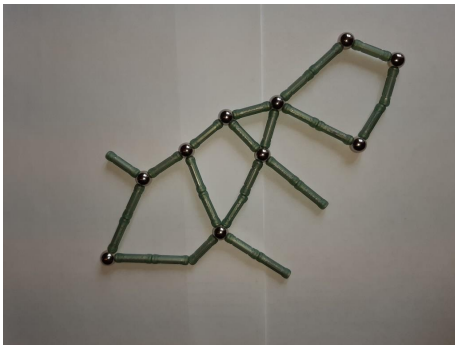


Рис. 1: Класс 1

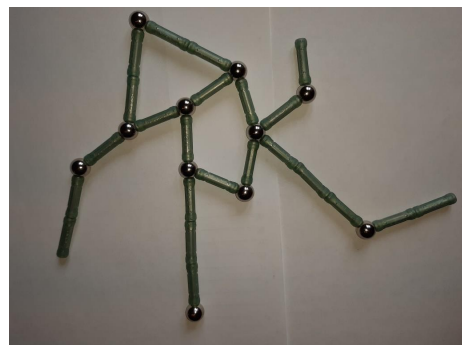
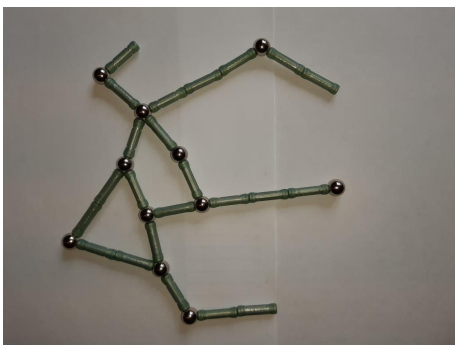


Рис. 2: Класс 2

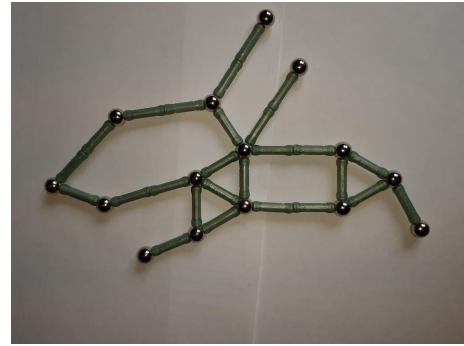
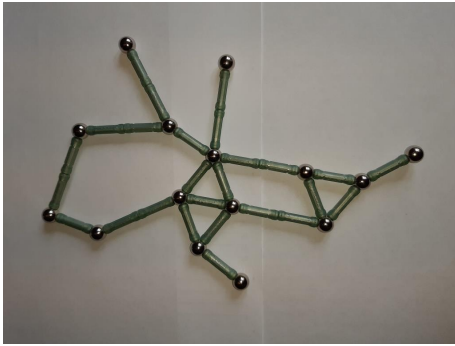


Рис. 3: Класс 3

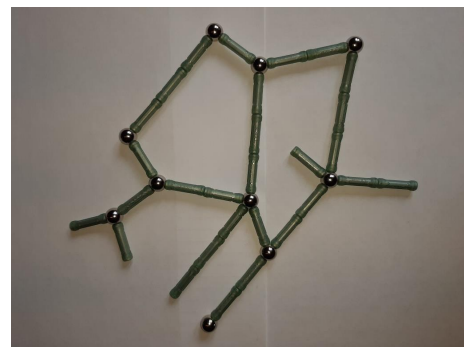
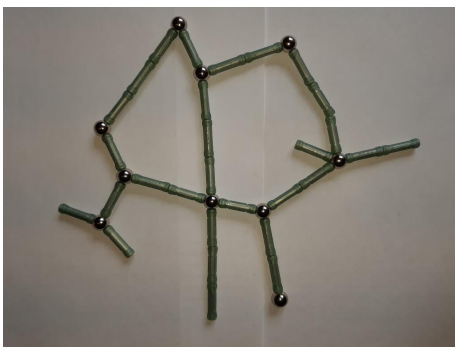


Рис. 4: Класс 4

## Метод решения и программная реализация

Для решения задачи использовались библиотеки `cv2` и `plantcv` для обработки изображений.

Данное на вход изображение сначала переводится в оттенки серого с помощью функции `cv.cvtColor`, а после производится бинаризация изображения с помощью функции `pcv.threshold.binary`. Ниже приведены два примера получившихся после этих операций изображений. Как видно, получившиеся изображения блеклые и не сплошные, а построенный по ним скелет имеет "неправильную" форму. Чтобы исправить этот эффект, к бинаризованным изображениям была применена дилатация с помощью функции `cv.dilate`, тем самым были невилированы неточности в изображениях графов.

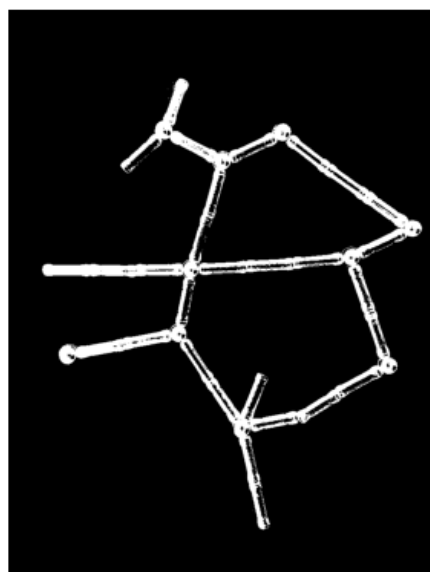
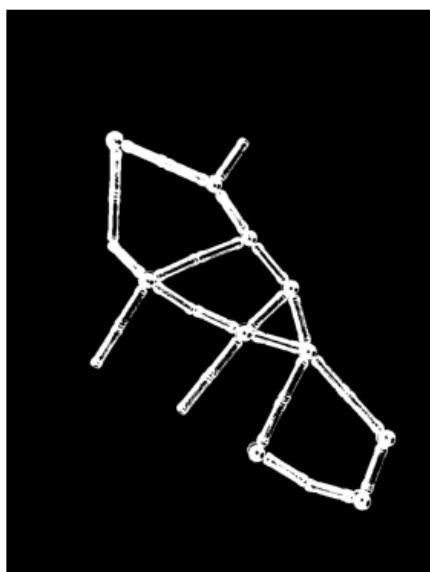


Рис. 5: Бинаризованные изображения

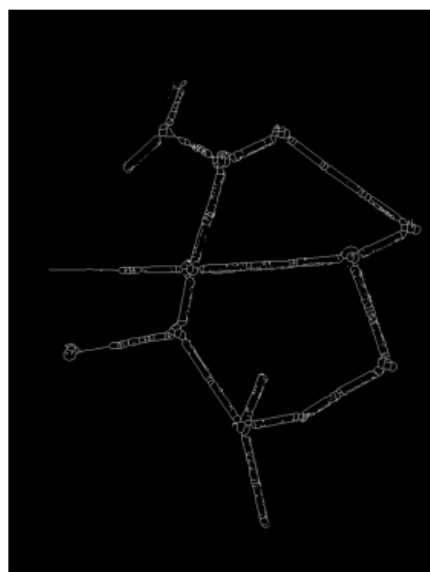
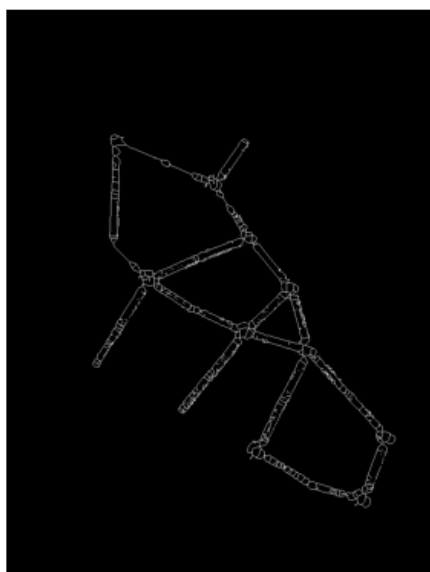


Рис. 6: Скелет, построенный по бинаризованным изображениям

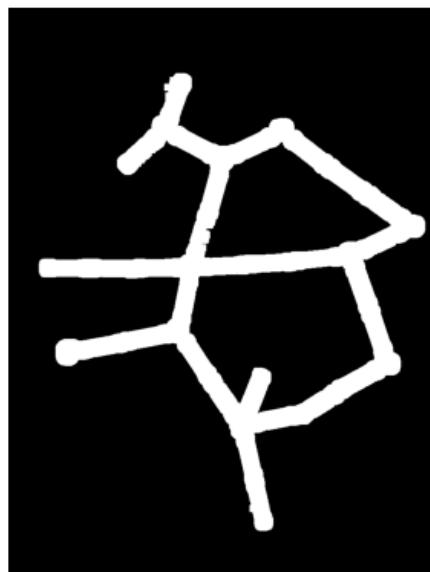


Рис. 7: Применение дилатации к бинаризованным изображениям

Для получившихся изображений был построен скелет с помощью функции `psv.morphology.skeletonize`

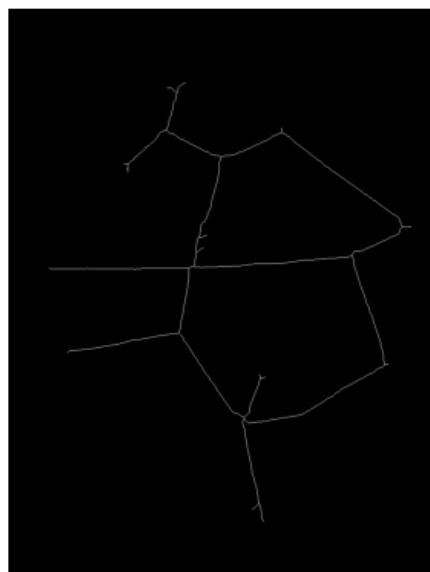
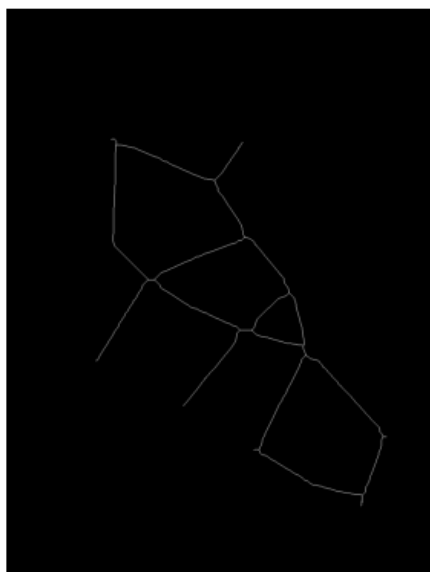


Рис. 8: Скелет изображений

Далее была применена операция прунинга для скелета, чтобы удалить неточности и сделать структуру скелета "более правильной". Для этого была использована функция `psv.morphology.prune`.

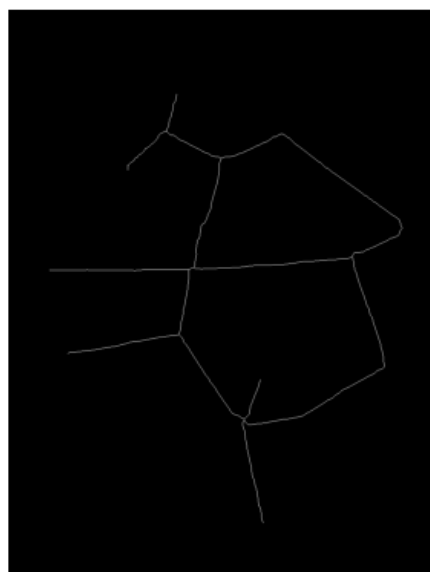
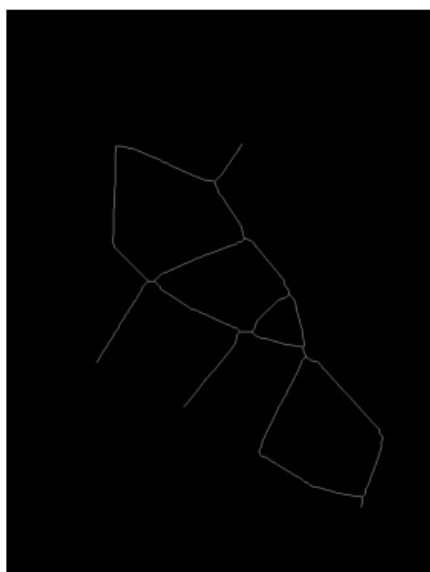


Рис. 9: Применение прунинга к скелетам изображений

Совместим скелет с исходным изображением, чтобы посмотреть на результат:

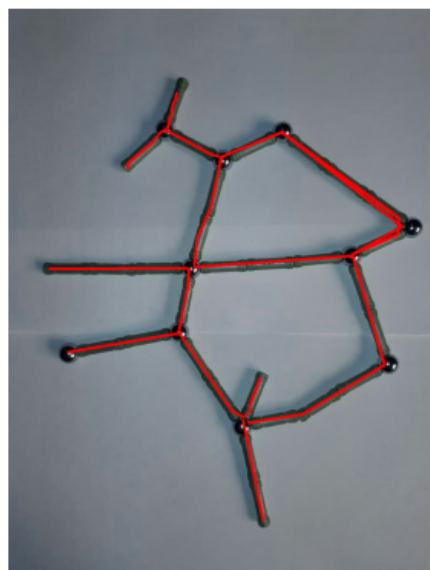
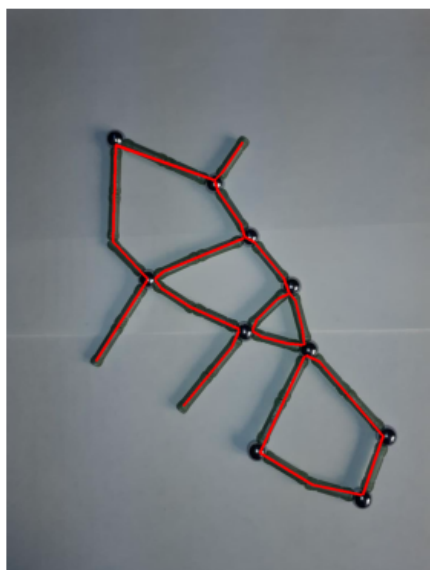


Рис. 10: Скелет на исходном изображении

Теперь необходимо выделить вершины построенного скелета. Для этого для каждой линии скелета сохранялась его максимальная и минимальная точка по координатам длины и ширины, а также исследовался угол наклона для каждой точки прямой. Если коэффициент прямой в определенной точке менял свой знак на противоположный, то такая точка сохранялась как вершина. Данный способ не является идеальным, и из-за него иногда в работе алгоритма появляются ошибки. Связано это с тем, что иногда точка не меняет направление

прямой, но при этом является вершиной, и наоборот, точка меняет направление прямой, но вершиной при этой не является.

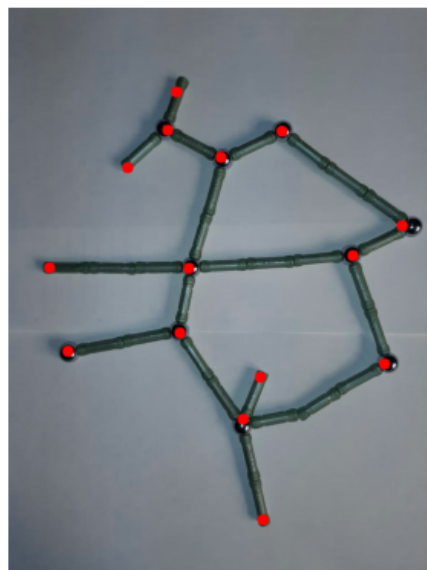
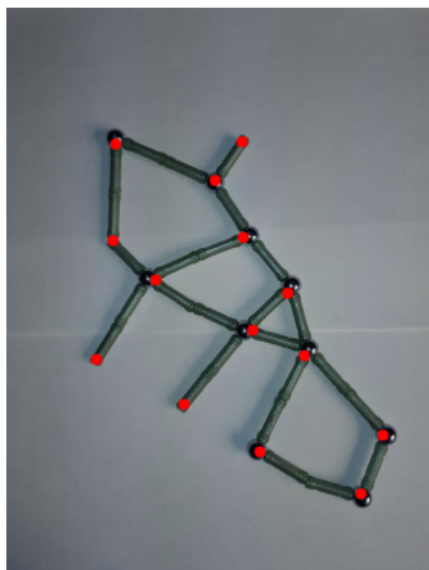


Рис. 11: Построенные вершины графа

Теперь необходимо посчитать степень каждой построенной вершины. Для этого заводится матрица смежности, и алгоритм проходится по каждой линии скелета. Если вершины являются смежными, то это сохраняется в матрицу смежности. По этой матрице строится вектор чисел, в котором хранится число вершин каждой степени. Построенную матрицу можно использовать и по-другому, так как она полностью описывает топологическую структуру графа.

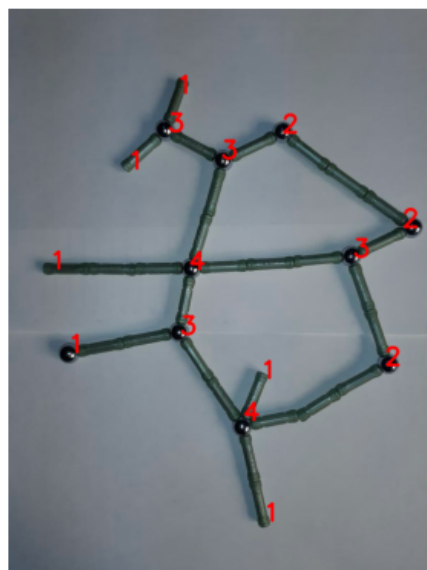
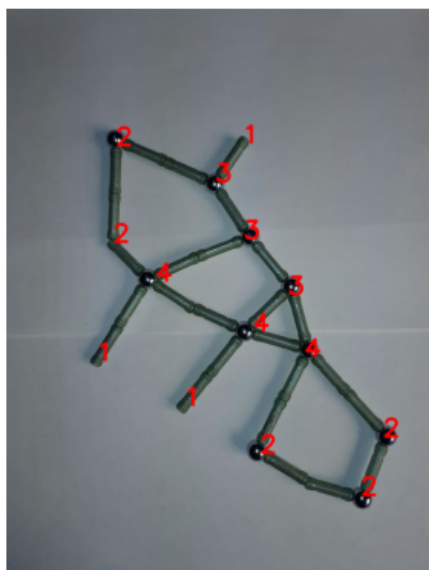


Рис. 12: Степени вершин графа



Классификация графа проводится по построенному вектору. Граф относится к тому классу, с которым у него больше всего совпадает вектор степеней вершин графа.

## Эксперименты

Алгоритм был протестирован на всех 8 изображениях, и на всех он показал правильный результат. Как уже упоминалось выше, алгоритм может неверно классифицировать какую-либо из вершин, но происходит это редко, и даже если алгоритм допускает ошибку, он верно классифицирует граф, поскольку классы графов отличаются более, чем в одной вершине.

## Выводы

Алгоритмы обработки изображений, и в частности алгоритм построения скелета изображения, позволили решить эту задачу: найти вершины графа и построить топологическую структуру графа.