

# Отчет о лабораторной работе № 2 «Изучение и освоение методов анализа формы объектов по изображениям».

317 группа, ММП ВМК МГУ.

Булкин Антон Павлович.

Апрель 2025.

## Содержание

<b>1</b>	<b>Постановка задачи</b>	<b>2</b>
<b>2</b>	<b>Описание данных</b>	<b>3</b>
<b>3</b>	<b>Описание используемых методов</b>	<b>3</b>
3.1	Гауссово размытие . . . . .	3
3.2	Бинаризация изображения . . . . .	4
3.3	Морфологические операции . . . . .	4
3.3.1	Closing . . . . .	4
3.3.2	Opening . . . . .	4
3.4	Преобразование расстояний . . . . .	4
3.5	Watershed . . . . .	5
3.6	Аппроксимация контуров методом Дугласа–Пекера . . . . .	5
3.7	Анализ кривизны контура . . . . .	6
3.8	Определение выпуклости . . . . .	6
<b>4</b>	<b>Описание программной реализации</b>	<b>6</b>
4.1	Основные модули и библиотеки . . . . .	6
4.2	Структура и логика программы . . . . .	7
4.2.1	Вспомогательные функции . . . . .	7
4.2.2	Класс CardsApp . . . . .	8
4.3	Особенности интерфейса и пользовательского взаимодействия . . . . .	9

## 1 Постановка задачи

Целью данной лабораторной работы является разработка и реализация программы для анализа формы объектов на цифровых изображениях, предназначенных для обработки карточек игрового набора «Геометрика».

Задача состоит в реализации следующих возможностей:

- **Ввод и отображение изображений:** Пользователь должен иметь возможность загрузить изображение карточек и увидеть его на экране в исходном виде.
- **Сегментация изображения:** Программа должна реализовывать сегментацию карточек на основе точечных и пространственных преобразований, что позволит выделить карточки из фона и друг от друга.
- **Поиск карточек на изображениях:** Программа должна находить все карточки, включая изолированные и частично перекрывающиеся друг с другом.
- **Выделение и распознавание фигур на карточках:** На найденных карточках программа должна определить тип фигуры (многоугольник или фигура с гладкой границей), количество вершин многоугольников и выпуклость многоугольников.

Результаты анализа изображений маркируются следующим образом:

- Карточки маркируются порядковым номером ( $N^{\circ}1$ ,  $N^{\circ}2$ , ...).
- Фигуры на карточках маркируются как:
  - $PnC$  — многоугольник с указанием числа вершин ( $n$ ) и выпуклости ( $C$ ).
  - $Pn$  — многоугольник без указания выпуклости.
  - $Smooth$  — фигура с гладкой границей.

Входные данные программы представлены в формате JPG, а результатом работы программы является исходное изображение с нанесённой маркировкой.

## 2 Описание данных

В качестве входных данных используются изображения карточек игрового набора «Геометрика», предоставленные в формате JPG. Изображения различаются по следующим признакам:

- **Фон изображения:**

- *Белый фон (#)*: Карточки размещены на простом однородном фоне, что облегчает сегментацию.
- *Пёстрый фон (ℰ)*: Карточки размещены на сложном, неоднородном фоне, что затрудняет процесс сегментации и распознавания.

- **Расположение карточек:**

- *Изолированные карточки*: Карточки не перекрывают друг друга, что упрощает их выделение.
- *Перекрывающиеся карточки*: Карточки частично наложены друг на друга, усложняя процесс выделения отдельных карточек.

- **Характеристики фигур на карточках:**

- *Тип фигуры*: Многоугольник или фигура с гладкой границей.
- *Количество вершин*: Для многоугольников количество вершин варьируется, что используется при маркировке.
- *Выпуклость*: Многоугольники могут быть выпуклыми или невыпуклыми, что также необходимо определить.

## 3 Описание используемых методов

В данной лабораторной работе были использованы классические методы обработки изображений и компьютерного зрения для решения задачи. Основные методы, применённые в реализации, рассмотрены ниже.

### 3.1 Гауссово размытие

Для подавления шума и уменьшения мелких дефектов изображения используется гауссово размытие, основанное на свёртке изображения с гауссовым ядром:

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right),$$

где  $\sigma$ — стандартное отклонение гауссианы, контролирующее степень размытия.

## 3.2 Бинаризация изображения

В задании используется пороговая бинаризация, определяемая формулой:

$$B(x, y) = \begin{cases} 255, & I(x, y) \leq T, \\ 0, & I(x, y) > T, \end{cases}$$

где  $I(x, y)$ — интенсивность пикселя серого изображения, а  $T$ — заданное пороговое значение, регулируемое пользователем.

В конкретной реализации применяется инвертированная бинаризация (*THRESH\_BINARY\_INV*), при которой объекты отображаются белым (значение 255), а фон чёрным (значение 0).

## 3.3 Морфологические операции

Используются морфологические операции для устранения шумов и разрывов на бинаризованном изображении:

### 3.3.1 Closing

Замыкание применяется для устранения мелких разрывов и дефектов на объектах, реализуется последовательным выполнением дилатации и эрозии:

$$I_{\text{closed}} = (I \oplus K) \ominus K,$$

где  $\oplus$ — дилатация,  $\ominus$ — эрозия,  $K$ — ядро.

### 3.3.2 Opening

Открытие удаляет мелкие объекты и шумы, сохраняя при этом основные формы объектов:

$$I_{\text{open}} = (I \ominus K) \oplus K.$$

## 3.4 Преобразование расстояний

Преобразование расстояний вычисляет расстояние от каждого пикселя объекта до ближайшего фонового пикселя, позволяя разделить перекрывающиеся объекты. Для каждого пикселя объекта:

$$D(x, y) = \min_{(x', y') \in \text{background}} \sqrt{(x - x')^2 + (y - y')^2}.$$

### 3.5 Watershed

Watershed используется для эффективного разделения соприкасающихся или перекрывающихся объектов. Алгоритм работает следующим образом:

- Создаётся карта маркеров, которые соответствуют центрам объектов.
- Маркеры и фоновые области рассматриваются как источники, из которых начинается условное «затопление» изображения.
- Границы раздела вод, возникающие при «затоплении», формируют чёткие контуры разделения объектов.

Алгоритм формально можно представить как задачу минимизации функции энергии, заданной градиентом изображения, и маркерными условиями:

$$E = \int_{\Omega} |\nabla I(x, y)| dx dy \rightarrow \min,$$

где  $\nabla I(x, y)$ — градиент исходного изображения,  $\Omega$ — область сегментации, заданная маркерами.

### 3.6 Аппроксимация контуров методом Дугласа–Пекера

Контуры объектов аппроксимируются многоугольниками с использованием метода Дугласа–Пекера, что позволяет упростить контур с минимальными потерями формы объекта:

$$P = \text{approxPolyDP}(C, \epsilon, \text{True}),$$

где:

- $C$ — исходный контур.
- $P$ — упрощённый многоугольник.
- $\epsilon$ — максимальное допустимое расстояние между исходным контуром и упрощённым многоугольником.

## 3.7 Анализ кривизны контура

Для различения многоугольников и гладких фигур используется анализ локальной кривизны контура. Кривизна рассчитывается на основе углов между соседними сегментами контура:

$$\kappa_i = \arccos \left( \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|} \right),$$

где  $\mathbf{v}_1, \mathbf{v}_2$  — векторы, образованные соседними точками контура. Количество выраженных локальных максимумов кривизны позволяет классифицировать фигуры.

## 3.8 Определение выпуклости

Выпуклость контура проверяется функцией, проверяющей для каждой точки контура выполнение условия выпуклости. Контур является выпуклым, если для любой его точки выполняется:

$$\forall i, \quad (p_{i+1} - p_i) \times (p_{i+2} - p_{i+1}) \geq 0,$$

где  $p_i$  — последовательные точки контура.

# 4 Описание программной реализации

Разработанная программа реализована на языке Python с использованием библиотек для обработки изображений и построения пользовательского интерфейса. Ниже подробно рассмотрены основные компоненты реализации.

## 4.1 Основные модули и библиотеки

В реализации использованы следующие Python-библиотеки и модули:

- **OpenCV (cv2):**

- Загрузка изображений из файлов.
- Преобразование изображений (перевод в оттенки серого, бинаризация, размытие).
- Морфологические преобразования (эрозия, дилатация, открытие, закрытие).
- Алгоритмы сегментации (Distance Transform, Watershed).

- Анализ и аппроксимация контуров.
- **NumPy:**
  - Выполнение математических операций и обработка массивов изображений.
- **Pillow (PIL):**
  - Конвертация изображений из формата OpenCV в формат, пригодный для отображения в интерфейсе Tkinter.
- **Tkinter:**
  - Создание графического пользовательского интерфейса.
  - Взаимодействие с пользователем через графические элементы - кнопки и ползунки.
  - Загрузка, отображение и сохранение изображений и результатов обработки.

## 4.2 Структура и логика программы

Программа организована в виде основного класса `CardsApp`, который управляет загрузкой, обработкой изображений и пользовательским интерфейсом, и набора вспомогательных функций, которые реализуют отдельные шаги обработки изображений.

### 4.2.1 Вспомогательные функции

Реализованные вспомогательные функции отвечают за конкретные задачи обработки изображений:

- `imread_unicode(path)` — Загрузка изображений, включая пути с unicode-символами.
- `preprocess_mask(img, bin_thr, closing_it)` — Предварительная обработка изображения, включая размытие, бинаризацию и морфологическое замыкание.
- `split_overlapping(mask, open_it, dilate_it, dist_frac)` — Разделение перекрывающихся карточек методом Watershed с предварительным преобразованием расстояний.

- `find_cards(img, mask, open_it, dilate_it, dist_frac)` — Поиск и фильтрация контуров карточек по площади.
- `curvature_peaks(contour, frac_thresh)` — Определение количества значимых пиков кривизны контура, используемое для классификации форм фигур на карточках.
- `annotate_shapes(...)` — Функция, маркирующая контуры карточек и классифицирующая фигуры на них, включая выделение многоугольников и фигур с гладкой границей.

#### 4.2.2 Класс CardsApp

Класс `CardsApp` реализует пользовательский интерфейс и управляет процессом взаимодействия пользователя с программой. Его структура включает следующие основные элементы и методы:

- **Инициализация интерфейса:**
  - Задаются параметры обработки изображений по умолчанию, с возможностью их изменения.
  - Создаются графические элементы интерфейса.
- **Выбор изображения:**
  - Пользователь выбирает файл изображения через диалоговое окно, после чего изображение отображается в интерфейсе.
- **Обработка изображения:**
  - Пользователь запускает процесс обработки изображения с текущими параметрами, после чего отображаются результаты сегментации и анализа.
- **Отображение результатов:**
  - Интерфейс отображает исходное изображение, бинарную маску, результаты разделения объектов и итоговое изображение с размеченными карточками и фигурами.
- **Сохранение результатов:**
  - Пользователь может сохранить результаты обработки в указанную им директорию.



## 4.3 Особенности интерфейса и пользовательского взаимодействия

Интерфейс программы выполнен на основе библиотеки Tkinter и обеспечивает следующие возможности:

- Настройка параметров обработки изображений с помощью удобных шкал и регуляторов, позволяющих подбирать оптимальные настройки под различные изображения.
- Отображение промежуточных и итоговых результатов в реальном времени для удобства анализа и отладки алгоритмов.
- Интуитивно понятная структура интерфейса, упрощающая взаимодействие с программой пользователям, не знакомым с деталями реализации алгоритмов.

## 4.4 Демонстрация работы программы

- Пользовательский интерфейс программы:

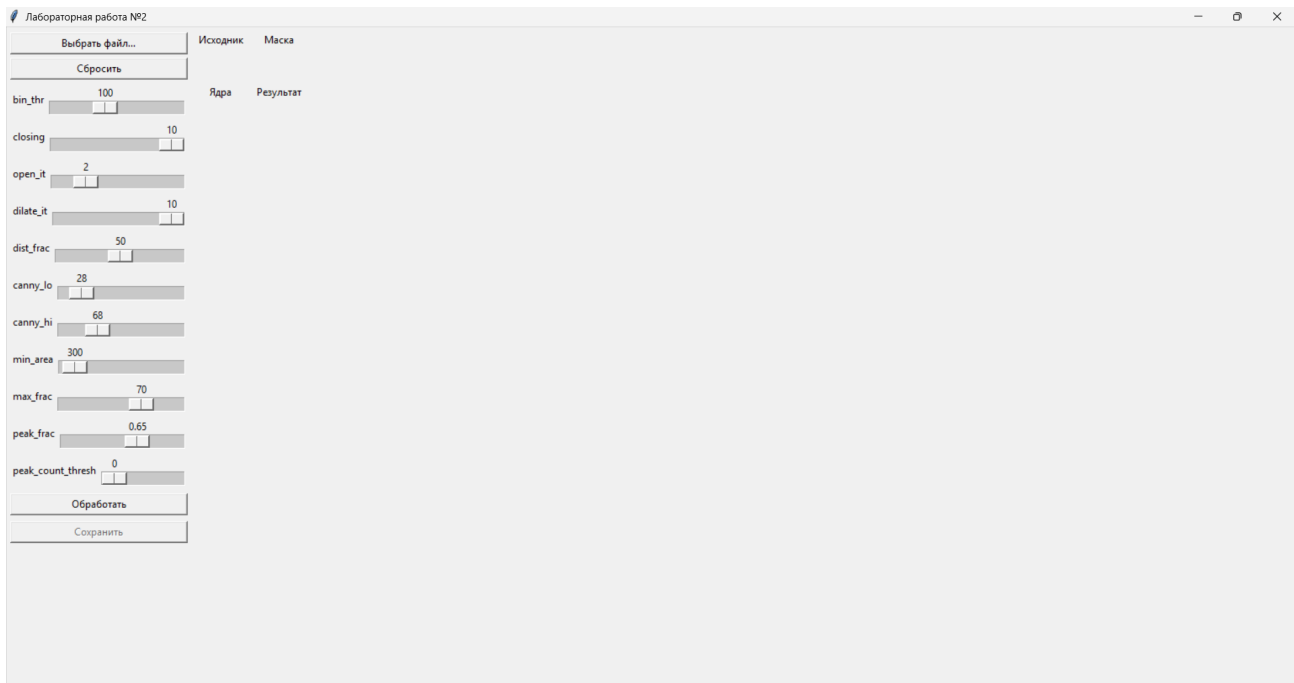


Рис. 1: Изначальный пользовательский интерфейс

- После добавления и обработки файла:

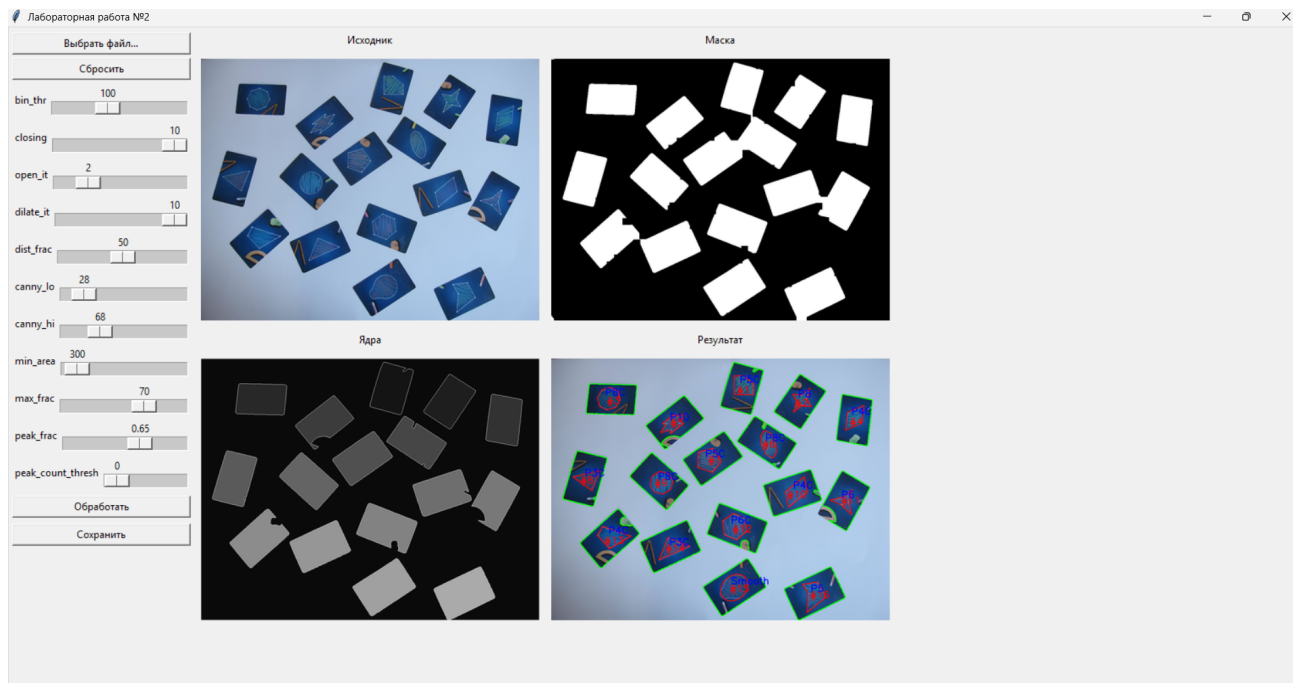


Рис. 2: Пользовательский интерфейс после выбора и обработки файла

### Пример демонстрационных изображений:

- Исходное изображение:



Рис. 3: Исходное изображение, выбранное пользователем

- Маска объектов:



Рис. 4: Бинарная маска после морфологической обработки

- Ядра:

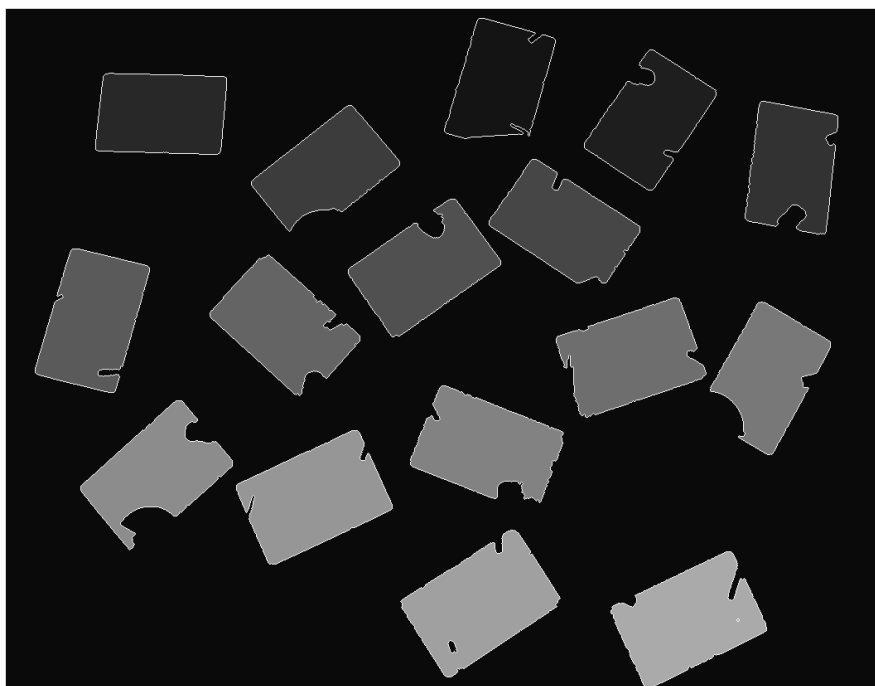


Рис. 5: Ядра для объектов

- **Итоговое изображение:**



Рис. 6: Итоговое изображение с выделенными контурами карточек и классифицированных фигур

## 5 Выводы

В рамках лабораторной работы была разработана и успешно реализована программа, предназначенная для анализа и сегментации изображений карточек игрового набора «Геометрика». По результатам выполненной работы можно сделать следующие выводы:

- **Эффективность методов:** Используемые методы бинаризации, морфологических операций и маркерной водораздельной сегментации обеспечивают высокое качество выделения карточек на изображениях. В частности, применение предварительной обработки (гауссово размытие, замыкание) позволяет значительно улучшить результаты сегментации за счёт уменьшения шума и выделения чётких контуров объектов. Однако, из-за разной яркости картинок, часто приходится подбирать трешхолды для различных параметров вручную, что снижает общую эффективность программы.
- **Распознавание фигур:** Реализованные алгоритмы анализа кривизны контура и аппроксимации контуров методом Дугласа–Пекера позволяют идентифицировать типы фигур (многоугольники и гладкие фигуры) и опреде-

лять характеристики многоугольников (количество вершин и выпуклость), но иногда алгоритм может также ошибаться, принимая за гладкие фигуры правильные многоугольники с большим количеством вершин (для исправления этого были добавлены элементы подбора краевых значений для более эффективной классификации фигур)

- **Адаптивность к различным условиям:** Разработанный интерфейс и предусмотренные программой настройки дают возможность оперативно подбирать параметры алгоритмов под конкретные изображения, что обеспечивает хорошую адаптацию алгоритма к изображениям разного уровня сложности и различным условиям освещения и фона.

Таким образом, по поставленной задаче была реализованна программа, предполагающая реализацию подбора параметров и решения поставленной задачи, а реализация продемонстрировала надёжность и высокую степень гибкости классических методов компьютерного зрения при решении практических задач обработки изображений.

## Список литературы

- [1] Гонсалес Р., Вудс Р., Цифровая обработка изображений.: Пер. с англ. - М.: Техносфера, 2012. - 1104 с. (3-е издание, дополненное и исправленное)