

Федеральное государственное автономное образовательное учреждение высшего образования
РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет искусственного интеллекта

Лабораторная работа №1

Тараканов Борис Александрович

Ст. билет: 1032259220

Группа ЗФИмд-01-25

*26 номер в журнале
12 вариант лабораторной*

8 октября 2025 г.

Задание 1

Вариант 12. Масштабирование → шумоподавление (Median для «соли/перца») → Otsu vs Adaptive Mean → closing → СК/свойства → подсчёт. Сравните медианный vs гауссовский фильтры перед порогованием.

Вначале я сгенерировал три изображения для лабораторной, используя уже реализованную функцию `generate_synthetic_random()`, изображения различаются по сложности:

```
img_easy,      gt_easy,      n_easy      = generate_synthetic_random(clutter=0.05,  
with_holes=False)  
img_medium,    gt_medium,    n_medium    = generate_synthetic_random(clutter=0.15,  
with_holes=True)  
img_hard,      gt_hard,     n_hard      = generate_synthetic_random(clutter=0.35,  
with_holes=True)
```

Эти изображения сохраняются в папку, ниже пример исходных изображений, которые будут фигурировать в последующих шагах (Рис. 1)

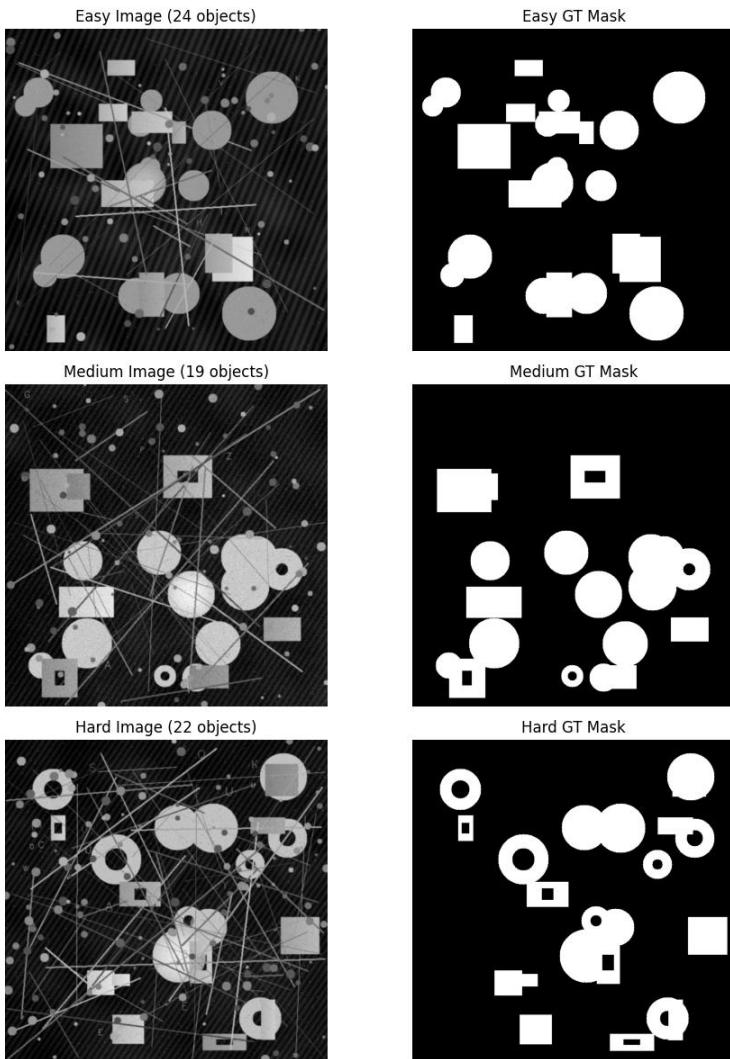


Рисунок 1 Сгенерированные изображения

После этого мне предстояло создать функцию обработки изображений, но для этого нужно было определиться с вариантами методов и параметрами для них.

Для этого я создал функцию `compare_methods()`, она предназначена для проведения сравнительного анализа различных комбинаций методов фильтрации, бинаризации и морфологической обработки изображений в задаче сегментации объектов.

На вход функция принимает исходное изображение, а также, при наличии, эталонную бинарную маску (ground truth), что даёт возможность количественно оценить точность сегментации и параметры для различных функций преобразования.

В процессе работы функция осуществляет масштабирование изображения, его фильтрацию с использованием медианного и гауссовского сглаживания, а затем выполняет бинаризацию двумя методами — глобальным (Otsu) и адаптивным (Adaptive Mean). После этого к результатам бинаризации применяется морфологическая операция замыкания (closing).

Для каждого варианта комбинации методов (Median + Otsu, Median + Adaptive, Gaussian + Otsu, Gaussian + Adaptive) функция производит подсчёт объектов двумя способами: методом А, основанным на анализе внешних и внутренних контуров и методом В, использующим эйлерову характеристику $\chi = C - H$, где С — число компонент связности, а Н — число отверстий.

Дополнительно при наличии эталонной маски функция вычисляет количественные метрики качества бинаризации: IoU (Intersection over Union), Precision (точность) и Recall (полнота). В результате выводятся различные варианты изображений и метрики (Рис. 2).

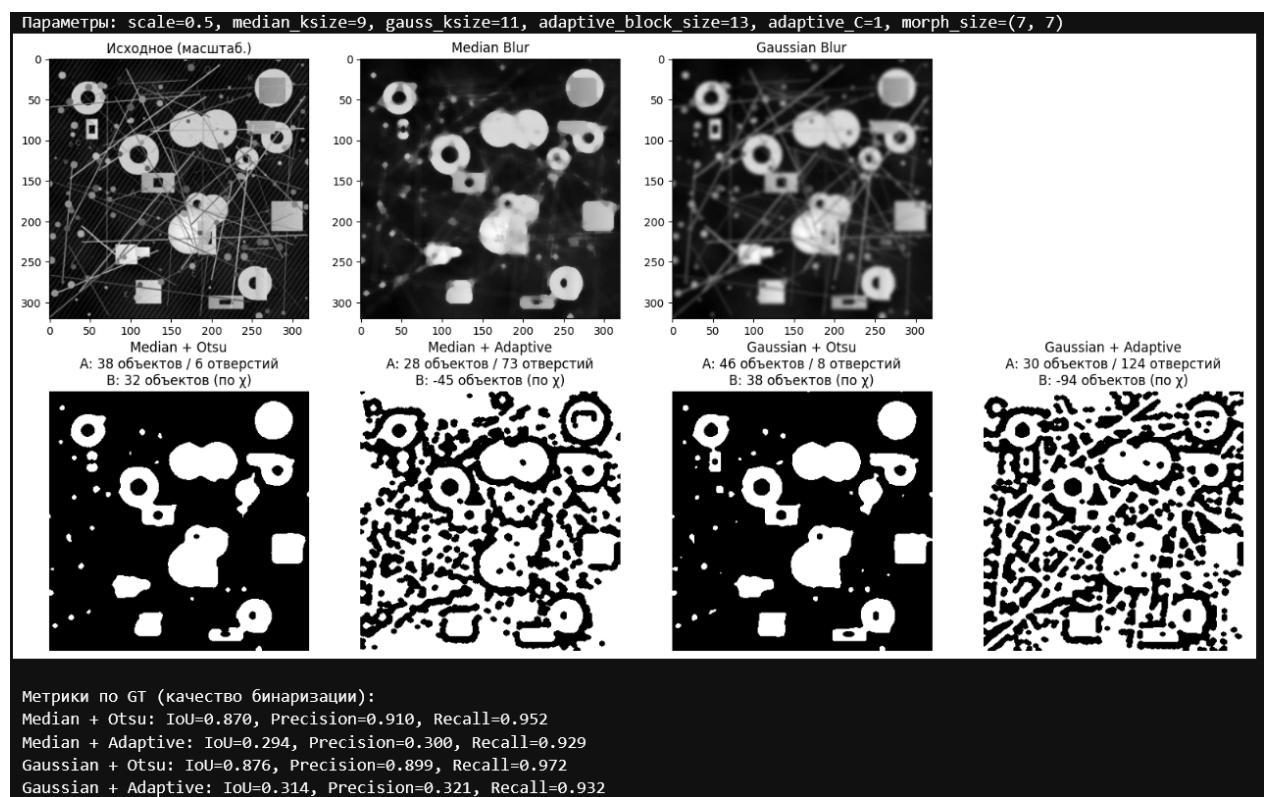


Рисунок 2: Один из результатов работы функции для подбора параметров

Выбор представленных методов и параметров обработки изображения обусловлен как требованиями варианта лабораторной работы, так и результатами предварительного сравнительного анализа, проведённого с использованием функции `compare_methods`. Целью подбора является обеспечение максимально устойчивого и корректного выделения объектов различной сложности.

На этапе масштабирования применён коэффициент `'scale=0.5'`, что позволяет уменьшить размер исходного изображения и, тем самым, снизить вычислительную нагрузку при сохранении ключевых структурных элементов сцены.

Для подавления шумов выбран метод гауссовского сглаживания с размером ядра `'gauss_ksize=11'` и дисперсией `'gauss_sigma=0'`. В отличие от медианного фильтра, гауссовское сглаживание лучше сохраняет общую структуру объектов и равномерно уменьшает влияние случайных шумов, не искажая форму границ (медианный фильтр хуже сохранял небольшие отверстия). Размер ядра подбирался эмпирически — значение 11 (при меньшем ядре на изображении оставалось больше полос шума).

Для этапа бинаризации выбран метод Otsu. В сравнении с адаптивным методом пороговой обработки, Otsu показал более стабильные результаты (адаптивный метод не удалось настроить так, чтобы он явно не выделял полосы на заднем фоне).

Применение морфологической операции замыкания (Closing) с использованием в качестве структурного элемента эллиптического ядра (`'morph_shape=cv.MORPH_ELLIPSE'`) размером `'(7, 7)'`. Эллиптическая форма обеспечивает более естественное воздействие на контуры объектов окружной формы, что характерно для синтетических данных, содержащих круги и прямоугольники с закруглёнными гранями. Размер ядра выбран исходя из необходимости подавления мелких дефектов при сохранении крупных структур.

Подсчёт числа объектов осуществляется методом A, основанным на анализе иерархии контуров. Этот метод позволяет отдельно учитывать количество внешних контуров (объектов) и внутренних (отверстий), что важно для корректной оценки изображений с элементами сложной формы. Данный подход признан более информативным и устойчивым к ошибкам морфологической обработки по сравнению с методом, основанным на эйлеровой характеристике.

Реализована маркировка связных компонент (8-связность), обеспечивающая количественную идентификацию и описание каждого объекта в терминах его площади, координат центра масс и параметров ограничивающего прямоугольника. Это позволяет формировать таблицу свойств областей, необходимую для последующего анализа и верификации корректности выделения сегментированных объектов.

Таким образом, выбранная комбинация методов — **масштабирование, гауссовское сглаживание, бинаризация по Otsu, морфологическое замыкание и подсчёт методом A** — обеспечивает наилучшее качество сегментации при работе с изображениями, содержащими шум, неоднородное освещение и объекты с

отверстиями. Подобранные параметры являются результатом экспериментального тестирования и обеспечивают устойчивость алгоритма к вариативности входных данных.

Далее подобранные параметры, реализованы в функции `pipeline_gaussian_otsu_A_final`. Она объединяет в единый алгоритм все этапы, необходимые для выделения и анализа объектов на изображении: масштабирование, фильтрацию, бинаризацию, морфологическую обработку, маркировку связных компонент и подсчёт объектов. Данный пайплайн обеспечивает автоматическое выполнение последовательности преобразований, сохранение итогового результата и вывод таблицы свойств областей, что позволяет проводить комплексный анализ изображений различной сложности. Ниже приведены примеры изображений полученных из функции (Рис. 3-5).



Задание 2

Описания изображений:

№ Имя файла Особенности съёмки

- | | | |
|---|-------|--------------------------------------|
| 1 | 1.jpg | Прямой взгляд, нейтральное освещение |
| 2 | 2.jpg | Взгляд с боку, нейтральное освещение |
| 3 | 3.jpg | Наличие очков |
| 4 | 4.jpg | Взгляд вниз |
| 5 | 5.jpg | Взгляд навер и влево |

Работа программы начинается с загрузки всех изображений из папки img_task_2. В данной папке находятся пять файлов в формате JPEG — снимки одного и того же человека, выполненные при различных условиях освещения, ракурсе и наличии шумов. Для каждого изображения осуществляется поочерёдная обработка с целью детектирования лица и определения 68 характерных точек.

Сначала изображение читается в память и преобразуется в оттенки серого, поскольку детектор лиц dlib работает устойчивее с одноканальными изображениями. После этого выполняется последовательность попыток обнаружения лица с различными параметрами. В каждой попытке может изменяться коэффициент upsample, определяющий, во сколько раз масштабируется изображение перед поиском лица. Это позволяет повысить чувствительность алгоритма к лицам малого размера. Дополнительно используется метод адаптивного выравнивания гистограммы (CLAHE), который улучшает контраст и делает возможной детекцию при сложных условиях освещения.

Для каждой попытки на изображении запускается детектор get_frontal_face_detector(). Если лицо обнаружено, выбирается прямоугольник с наибольшей площадью, так как именно он, как правило, соответствует основному объекту — лицу человека. Далее к этому прямоугольнику применяется предсказатель формы (shape_predictor_68_face_landmarks.dat), который определяет координаты 68 характерных точек. Если все точки были успешно найдены, процесс детекции для данного изображения считается успешным, и программа переходит к этапу визуализации.

На исходное изображение наносится прямоугольник, ограничивающий область лица, а также зелёные точки в позициях, соответствующих landmarks. Полученное изображение сохраняется в отдельную папку results_task_2, где для каждого исходного файла создаётся копия с нанесённой разметкой. Дополнительно в консоли фиксируется информация о параметрах успешной попытки (значение upsample, использование CLAHE) и статус выполнения.

Если ни в одной из попыток лицо не было найдено, программа регистрирует неудачу для данного файла и переходит к следующему изображению. В результате после выполнения программы для каждого кадра

формируется запись о статусе обработки — успешной или неуспешной, а также сохраняются визуализации для дальнейшего анализа качества локализации.

На одном из изображений (Рис. 6), мое лицо повернуто вверх и находится в сильном ракурсе, при котором изменяются видимые пропорции и нарушается симметрия. Алгоритм `dlib` не смог корректно определить область лица и зафиксировал прямоугольник детекции в нижней части изображения, ошибочно выделив участок одежды.

Такое поведение связано с тем, что модель `get_frontal_face_detector()` обучена преимущественно на фронтальных изображениях и не обладает высокой устойчивостью к большим углам поворота головы и перспективным искажениям. Для исправления подобных ошибок рекомендуется применять дополнительную фильтрацию по соотношению сторон области детекции, ограничивать поиск по высоте кадра или использовать более универсальные модели, например, CNN-детекторы лиц или архитектуры MTCNN / RetinaFace, способные работать с ракурсами.



Рисунок 5: Плохая детекция

На остальных изображениях где мое лицо не на столько повернуто, вне зависимости от наличия очков, алгоритм достаточно точно расставил все точки (Рис. 6)



Рисунок 6: Хорошая детекция

Результаты показали, что алгоритм демонстрирует высокую точность при анализе фронтальных изображений с равномерным освещением: все 68 точек корректно располагаются на анатомических ориентирах лица, что обеспечивает достоверность разметки. При умеренных изменениях наклона головы или наличии очков качество локализации остаётся удовлетворительным, возможны незначительные смещения отдельных точек в областях бликов или перекрытий.

Однако при существенном изменении ракурса, особенно при сильном повороте или запрокидывании головы, алгоритм dlib может ошибочно определять положение лица либо терять часть ключевых точек. На одном из снимков наблюдается явный пример некорректной детекции, когда область лица была ошибочно выделена на одежде. Такие ситуации объясняются ограничениями используемой модели, рассчитанной в первую очередь на фронтальные изображения.

Ссылка на исходный код:

https://github.com/4ebupelinka/CV_master_degree/blob/main/Lab_1/lab1_binary_images.ipynb