

Отчет о лабораторной работе № 1

«Обработка и сегментация изображений на примере подсчёта куриных яиц».

317 группа, ММП ВМК МГУ.

Булкин Антон Павлович.

Март 2025.

Содержание

1 Постановка задачи	2
2 Описание данных	2
3 Описание используемых методов	3
3.1 Цветовые пространства	3
3.2 CLAHE (Contrast Limited Adaptive Histogram Equalization)	4
3.3 Медианный фильтр	4
3.4 Бинаризация изображения (пороговая фильтрация)	4
3.5 Морфологические операции	4
3.6 Удаление малых компонент связности	5
3.7 Обнаружение и анализ контуров	5
4 Описание программной реализации	5
4.1 Основные модули и библиотеки	5
4.2 Структура и логика программы	6
4.2.1 Функция <code>count_eggs_with_params</code>	6
4.2.2 Класс <code>EggDetectorApp</code>	7
4.3 Демонстрация работы программы	8
5 Выводы	12

1 Постановка задачи

Целью данной лабораторной работы является разработка программы для работы с фотографиями, ориентированной на сегментацию объектов. Основное требование заключается в разделении изображения на отдельные объекты для последующего анализа. Задача включает следующие ключевые пункты:

- **Ввод и отображение изображений:** Пользователь должен иметь возможность выбрать файл с изображением, загрузить его в программу и увидеть исходное изображение на экране.
- **Сегментация изображения:** Программа должна реализовывать методы точечных и пространственных преобразований, позволяющих выделить объекты на изображении. Это включает применение алгоритмов, позволяющих различать объекты по границам, интенсивности цвета или другим характеристикам.
- **Бинаризация и подсчёт объектов:** После сегментации изображений необходимо выполнить их бинаризацию, т.е. преобразовать изображение в форму, где выделенные объекты представлены в одном цветовом тоне, а фон — в другом. На основе бинаризованного изображения программа должна подсчитать количество яиц, различая их по цвету.
- **Интерактивный режим работы:** Пользователь должен иметь возможность последовательно выбирать изображение, запускать алгоритмы преобразования, наблюдать за промежуточными результатами и получать итоговые данные о количестве объектов.
- **Адаптивность к сложности изображений:** Исходя из сложности фоновой подложки, изображения классифицируются на три уровня:
 - **Beginner:** Яйца на однородном фоне.
 - **Intermediate:** Изображения с яйцами, расположенными на фоне с незначительной неоднородностью.
 - **Expert:** Сложные изображения, где яйца находятся на ярком, пестром фоне.

2 Описание данных

В качестве входных данных используются цифровые изображения, на которых изображены куриные яйца, отличающиеся по цвету и размещенные на различных типах подложек. Подробное описание данных включает:

- **Характеристика изображений:**
 - Изображения представляют собой фотографии, содержащие объекты (яйца) различной окраски.
 - Фоновая подложка может быть как однородной, так и неоднородной, что обуславливает сложность задачи сегментации.
- **Классификация по сложности:**
 - **Beginner:** Изображения с однородным фоном. Данные являются наиболее простыми для сегментации, так как объекты легко выделяются благодаря высокой контрастности с фоном.
 - **Intermediate:** Изображения, где фон представляет собой линолеум небольшой яркости, что создает незначительные трудности при выделении объектов.
 - **Expert:** Изображения с ярким, пестрым фоном, где объекты слабо контрастируют с окружающей областью.

3 Описание используемых методов

В реализованной программе для сегментации и подсчёта объектов разного цвета (куриных яиц) на изображениях были использованы следующие методы компьютерного зрения:

3.1 Цветовые пространства

Использовались два цветовых пространства:

- **RGB** — модель цветового пространства, основанная на смешивании красного (R), зелёного (G) и синего (B) цветов. Используется для выделения белых объектов с помощью пороговой бинаризации.
- **LAB** — цветовое пространство, в котором цвет представлен компонентами яркости (L) и двух цветовых каналов (a и b). Это пространство эффективно разделяет освещённость и цветовую информацию и подходит для выделения красных объектов.

Преобразование из RGB в LAB выполняется по следующим формулам:

$$L = 116 \cdot f(Y/Y_n) - 16, \quad a = 500 \cdot [f(X/X_n) - f(Y/Y_n)], \quad b = 200 \cdot [f(Y/Y_n) - f(Z/Z_n)],$$

где функция $f(t)$ определяется как:

$$f(t) = \begin{cases} t^{1/3}, & t > (6/29)^3 \\ \frac{1}{3} \left(\frac{29}{6}\right)^2 t + \frac{4}{29}, & t \leq (6/29)^3 \end{cases}$$

3.2 CLAHE (Contrast Limited Adaptive Histogram Equalization)

CLAHE — метод улучшения контраста изображения, ограничивающий амплитуду гистограммы для предотвращения усиления шума. Для каждого участка изображения гистограмма модифицируется индивидуально. Формула для CLAHE не имеет единого аналитического выражения, поскольку это адаптивный и локализованный метод, но суть его применения заключается в преобразовании яркостного канала (L-канал) изображения в пространстве LAB.

3.3 Медианный фильтр

Медианный фильтр заменяет каждое значение пикселя медианой значений пикселей, находящихся в его окрестности:

$$I(x, y) = \text{median}\{I(x+i, y+j) | (i, j) \in \Omega\},$$

где Ω — окрестность пикселя.

3.4 Бинаризация изображения (пороговая фильтрация)

Используется пороговая бинаризация, формально описываемая следующим образом:

$$B(x, y) = \begin{cases} 255, & \text{если } L \leq I(x, y) \leq U \\ 0, & \text{иначе} \end{cases},$$

где L и U — нижний и верхний пороги бинаризации.

3.5 Морфологические операции

Используются морфологические операции с ядром K :

Открытие (Opening):

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

Закрытие (Closing):

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

где \oplus — дилатация, \ominus — эрозия.

3.6 Удаление малых компонент связности

Метод основан на анализе связных компонент бинарного изображения:

$$\text{CC} = \{C_i | \forall i, \text{Area}(C_i) \geq S_{\min}\},$$

где $\text{Area}(C_i)$ — площадь компоненты, а S_{\min} — минимальный размер компоненты.

3.7 Обнаружение и анализ контуров

Контуры выделяются по бинарному изображению. Контуры фильтруются по следующим критериям:

Площадь контура:

$$S_{\min} \leq \text{Area}(C) \leq S_{\max}$$

Соотношение сторон (Aspect Ratio):

$$\text{Aspect Ratio} = \frac{w}{h}, \quad AR_{\min} \leq \frac{w}{h} \leq AR_{\max}$$

Циркулярность (Circularity):

$$\text{Circularity} = \frac{4\pi \cdot \text{Area}(C)}{\text{Perimeter}(C)^2}, \quad C_{\min} \leq \text{Circularity} \leq C_{\max}$$

4 Описание программной реализации

Данная программа реализована на языке Python с использованием библиотек OpenCV, Tkinter, NumPy, Pillow и Matplotlib. Программа предназначена для детекции и сегментации объектов (куриных яиц) на изображениях, разделяя их по цвету (белые и красные) и предоставляя пользователю возможность настраивать параметры обработки через графический интерфейс. Далее представлено подробное описание каждого компонента реализации.

4.1 Основные модули и библиотеки

- **Tkinter:** Используется для построения графического интерфейса, позволяющего:
 - Выбирать изображение через диалоговое окно.
 - Настраивать параметры обработки (например, пороговые значения, параметры морфологических операций) с помощью шкал и спинбоксов.

- Отображать промежуточные результаты и итоговое изображение с выделенными контурами.
- **OpenCV:** Отвечает за выполнение основных операций обработки изображений:
 - Загрузка изображений из файлов.
 - Преобразование цветовых пространств (BGR → RGB, RGB → LAB).
 - Создание бинарных масок для сегментации.
 - Применение морфологических операций (открытие, закрытие, дилатация) и размытия для очистки изображений.
 - Выделение контуров и их фильтрация по площади, отношению сторон и круговости.
- **NumPy:** Применяется для работы с массивами данных и математическими операциями над изображениями.
- **Pillow (PIL):** Используется для конвертации изображений из формата OpenCV в формат, пригодный для отображения в Tkinter.

4.2 Структура и логика программы

Программа состоит из двух основных частей: функции для обработки изображений и класса приложения, реализующего графический интерфейс.

4.2.1 Функция count_eggs_with_params

Эта функция выполняет последовательную обработку изображения и включает следующие этапы:

1. Загрузка и конвертация изображения:

- Изображение загружается с использованием OpenCV.
- Конвертируется из BGR в RGB для корректного отображения.

2. Предобработка изображения:

- Для выделения белых объектов выполняется преобразование в цветовое пространство LAB.
- Применяется метод CLAHE (адаптивное гистограммное выравнивание) для улучшения контраста.

- Применяются медианное размытие и дилатация для устранения мелких шумов.

3. Создание бинарных масок:

- Для белых объектов используется диапазон значений в RGB на предобработанном изображении.
- Для красных объектов применяется пороговое выделение в цветовой модели LAB.

4. Морфологическая обработка и сглаживание:

- Применяются операции открытия и закрытия для улучшения качества бинарных масок.
- Выполняется размытие масок для устранения резких переходов и устранения шумов.

5. Удаление мелких компонентов:

- Функция `remove_small_components` удаляет объекты, площадь которых меньше заданного порога, что позволяет исключить шумы.

6. Выделение и фильтрация контуров:

- Контуры объектов выделяются с помощью функции `findContours`.
- Проводится фильтрация контуров по площади (с учетом минимальных и максимальных порогов), отношению сторон и круговости.
- Исключаются контуры, касающиеся границ изображения, чтобы избежать ошибок при подсчёте.

7. Визуализация результатов:

- Найденные контуры отображаются на исходном изображении (белые контуры выделяются одним цветом, красные — другим).
- Функция возвращает итоговое изображение, статистику по количеству объектов и промежуточные маски.

4.2.2 Класс EggDetectorApp

Класс `EggDetectorApp` реализует графический интерфейс пользователя и управляет процессом обработки изображений. Основные элементы класса:

1. Инициализация интерфейса:

- Окно приложения создается с помощью Tkinter.
- Интерфейс делится на две части: левая панель для настройки параметров и правая для отображения результатов.

2. Элементы управления:

- **Параметры обработки:** Шкалы, спинбоксы и кнопки позволяют пользователю настраивать пороговые значения для сегментации, параметры морфологических операций и другие ключевые настройки.
- **Выбор изображения:** Кнопка для вызова диалогового окна выбора файла позволяет загрузить изображение для дальнейшей обработки.

3. Обработка изображения:

- При нажатии на кнопку «Обработать» вызывается функция `count_eggs_with_p` с текущими параметрами, введенными пользователем.
- Результаты обработки (исходное изображение, маски белых и красных объектов, итоговое изображение с выделенными контурами) отображаются в правой части окна.

4. Сохранение результатов:

- Реализована возможность сохранения итоговых изображений в выбранную пользователем директорию для последующего анализа.

5. Конвертация и отображение изображений:

- Функция `cv2_to_pil` конвертирует изображения из формата OpenCV в формат PIL для корректного отображения в Tkinter.
- Функция `show_image_on_label` обеспечивает масштабирование и размещение изображений в отведенных для них областях интерфейса.

4.3 Демонстрация работы программы

- Пользовательский интерфейс программы:

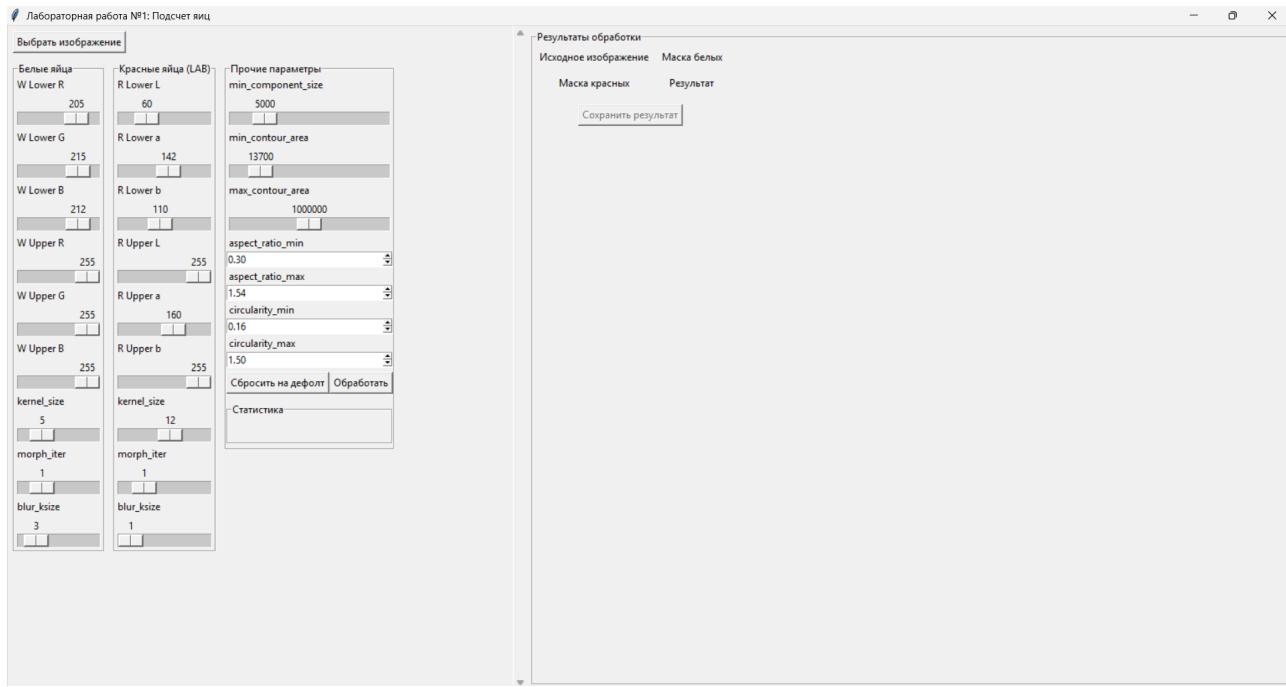


Рис. 1: Изначальный пользовательский интерфейс

- После добавления файла:

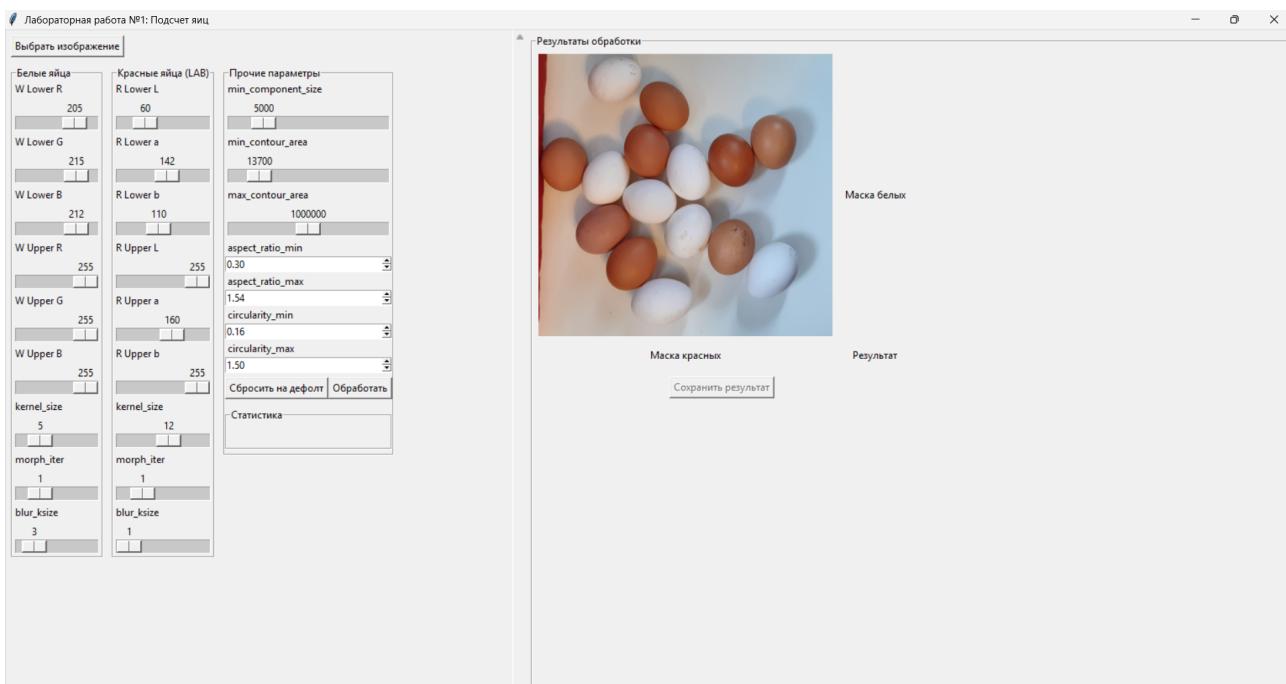


Рис. 2: Пользовательский интерфейс после выбора файла

- После обработки файла:

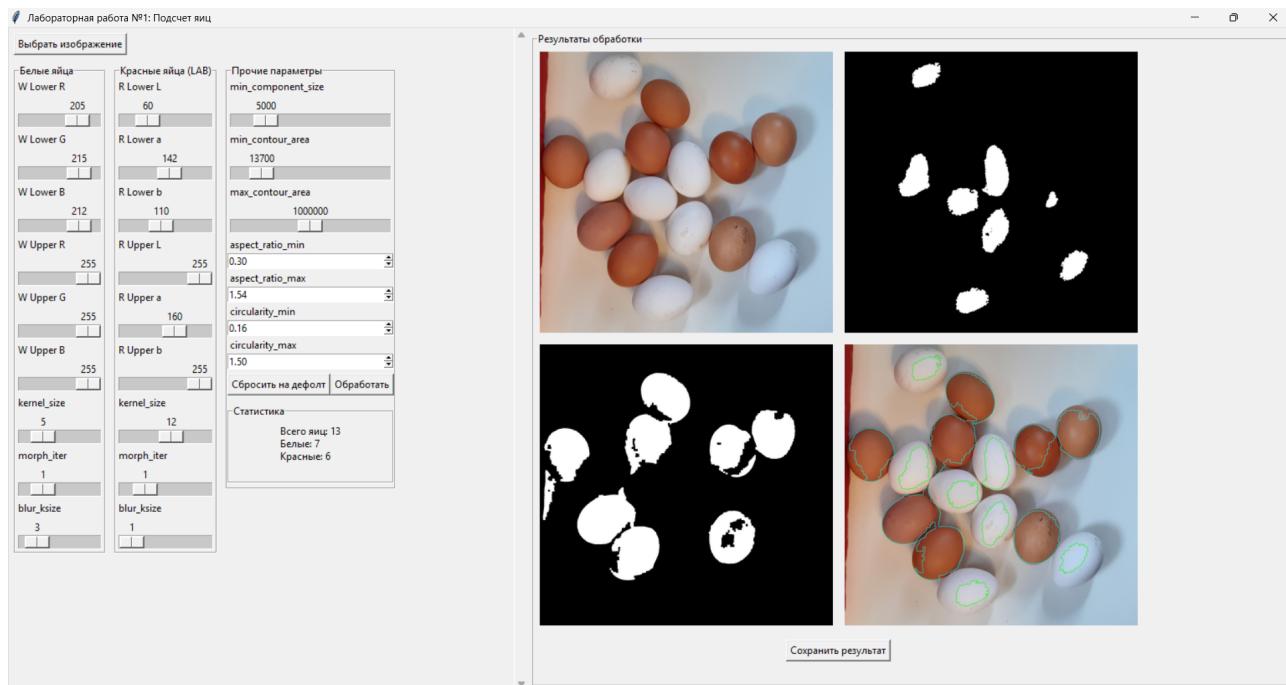


Рис. 3: Пользовательский интерфейс после обработки файла

Пример демонстрационных изображений:

- Исходное изображение:



Рис. 4: Исходное изображение, выбранное пользователем

- Маска белых объектов:

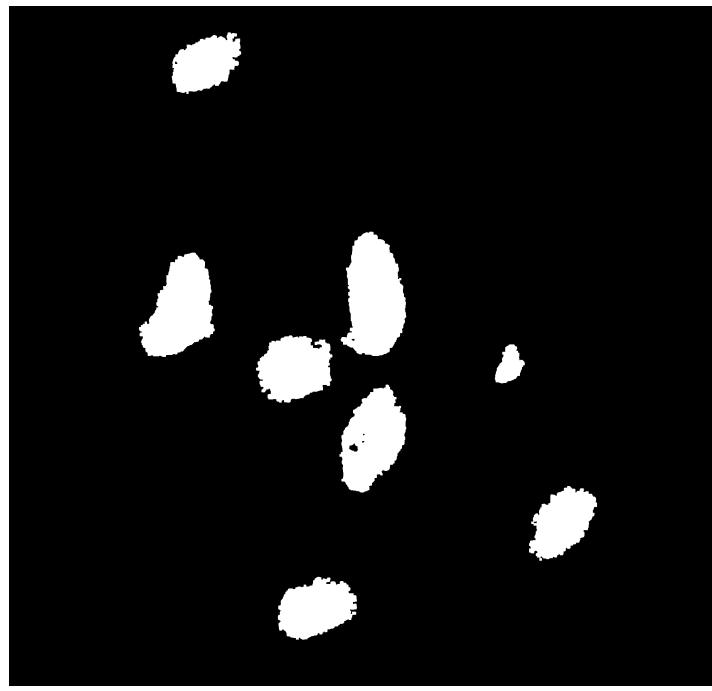


Рис. 5: Бинарная маска для белых объектов после морфологической обработки

- **Маска красных объектов:**

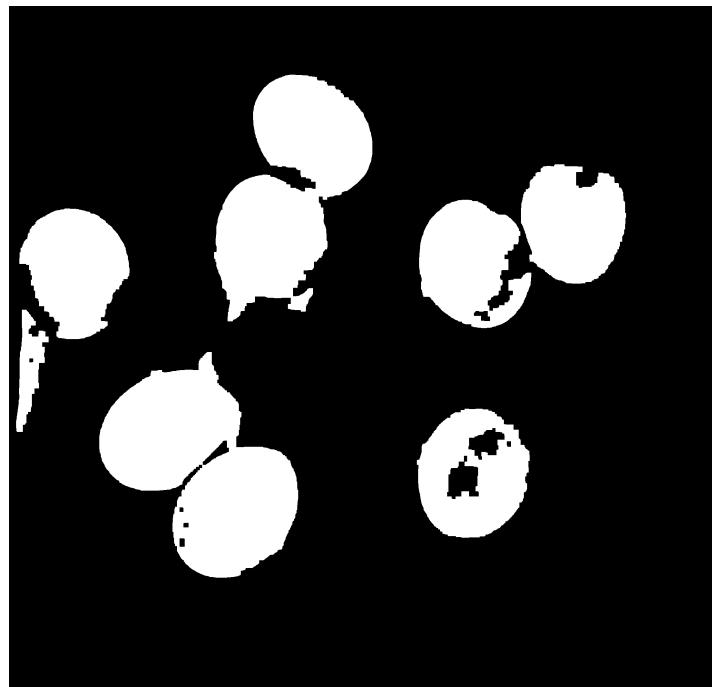


Рис. 6: Бинарная маска для красных объектов

- **Итоговое изображение:**

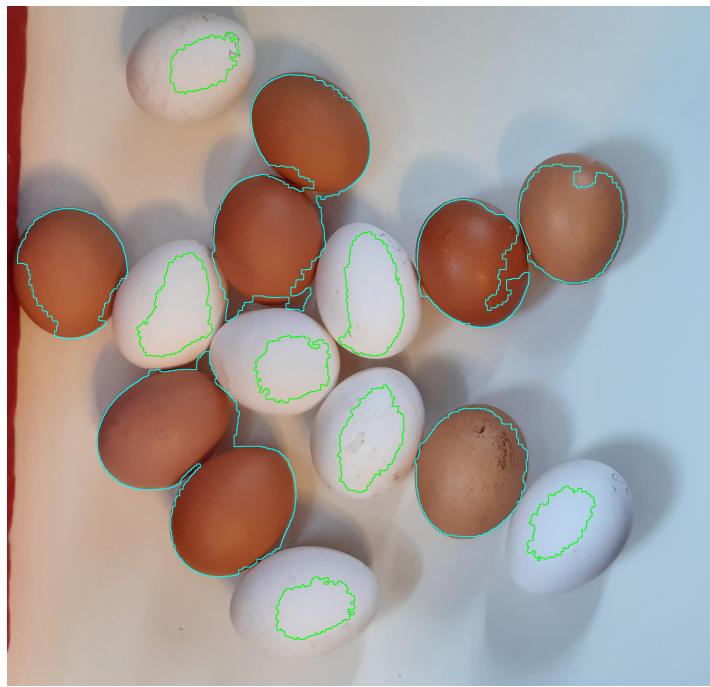


Рис. 7: Итоговое изображение с выделенными контурами обнаруженных яиц

5 Выводы

В результате выполненной лабораторной работы была разработана программа для сегментации и детекции объектов на изображениях, направленная на выделение куриных яиц по их цвету. Полученные результаты позволяют сделать следующие выводы:

- **Эффективность алгоритма:** Реализованные методы предварительной обработки, сегментации и фильтрации контуров продемонстрировали высокую точность при выделении объектов на изображениях с однородным фоном. Применение морфологических операций и адаптивной бинаризации позволило существенно снизить влияние шума и мелких артефактов.
- **Адаптивность к различным условиям:** Программа предусматривает возможность работы с изображениями различной сложности (классы Beginner, Intermediate и дополнительно Expert). Интерактивный режим настройки параметров позволяет пользователю оптимизировать процесс обработки под конкретные условия и особенности исходных изображений. Однако, из-за различной яркости изображений и различных шумов, не было возможно создать унифицированную систему threshold-ов, из-за чего часто приходится слегка изменять параметры для нахождения всех целевых сегментов.
- **Интерактивность и удобство использования:** Графический интерфейс,

реализованный на основе Tkinter, обеспечивает простой и наглядный механизм управления процессом обработки, а также позволяет в реальном времени просматривать промежуточные результаты. Это существенно упрощает диагностику и корректировку алгоритма.

- **Потенциал для дальнейшего развития:** Дальнейшие улучшения могут быть направлены на автоматическую адаптацию параметров обработки в зависимости от характеристик входного изображения, а также на расширение функционала для работы с изображениями более высокой сложности.

Таким образом, реализованное решение полностью соответствует поставленной задаче и демонстрирует универсальный и надежный подход к проблеме сегментации изображений, что подтверждается как теоретическим обоснованием, так и практическим применением.

Список литературы

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