



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,

информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №1

«Сборка и установка библиотеки OpenCV. Использование библиотеки в среде
Microsoft Visual Studio»

ДИСЦИПЛИНА: «Программные системы распознавания и обработки
информации»

Выполнил: студент гр. ИУК4-31М _____ (Сафронов Н.С,)
(подпись) (Ф.И.О.)

Проверил: _____ (Гагарин Ю.Е.)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

Цель:

Рассмотреть технические этапы подготовки инфраструктуры и продемонстрировать использование базовых функций библиотеки OpenCV на простых практических примерах.

Задачи:

1. Сборка и установка библиотеки OpenCV с использованием инсталлятора и из исходных кодов.
2. Настройка среды Microsoft Visual Studio с целью использования библиотеки при разработке C/C++ приложений.
3. Разработка приложения, демонстрирующего применение некоторых базовых операций обработки изображений

Задание

1. Модифицируйте приложение для поиска контуров объекта так, чтобы результирующее изображение сохранялось в файл.
2. Модифицируйте приложение для поиска контуров объекта так, чтобы отображалось изображение, конвертированное в оттенки серого, и бинаризованное изображение, т.е. то, что показано на рис. 17.
3. Как будет изменяться результат работы приложения для определения контуров, если изменить способ восстановления контуров? Проведите эксперименты со всеми возможными значениями параметра `mode` в функции `findContours`.
4. Как будет изменяться результат работы приложения для определения контуров, если изменить способ восстановления контуров? Проведите эксперименты со всеми возможными значениями параметра `method` в функции `findContours`.

5. Модифицируйте результаты задания 2 так, чтобы изображение, конвертированное в оттенки серого, и бинаризованное изображение отображались в одном окне.
6. Модифицируйте результаты задания 3 так, чтобы все изображения отображались в одном окне.
7. Модифицируйте результаты задания 4 так, чтобы все изображения отображались в одном окне.
8. Сохраните результаты задания 7 в файл.

Результаты выполнения работы



Рисунок 1 - Оригинальное изображение

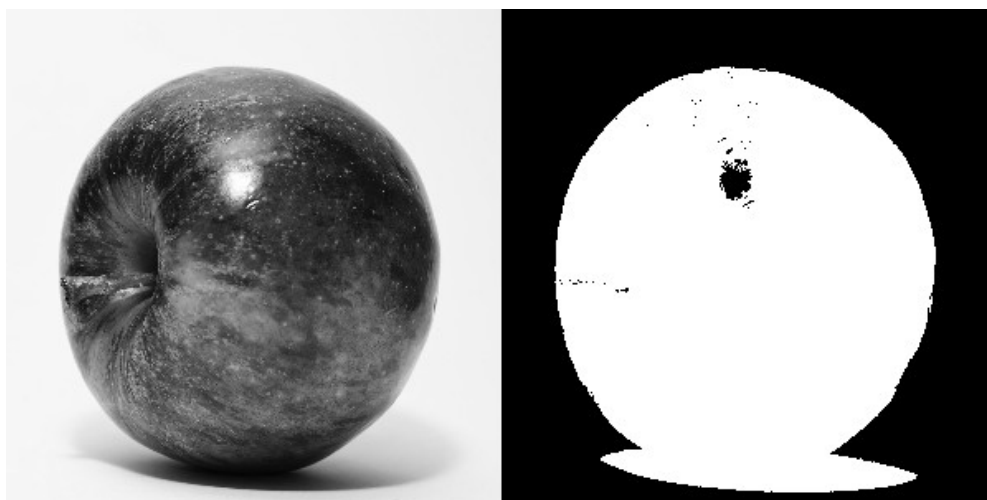


Рисунок 2 - Изображение в оттенках серого и его бинаризация (результат задания 5)

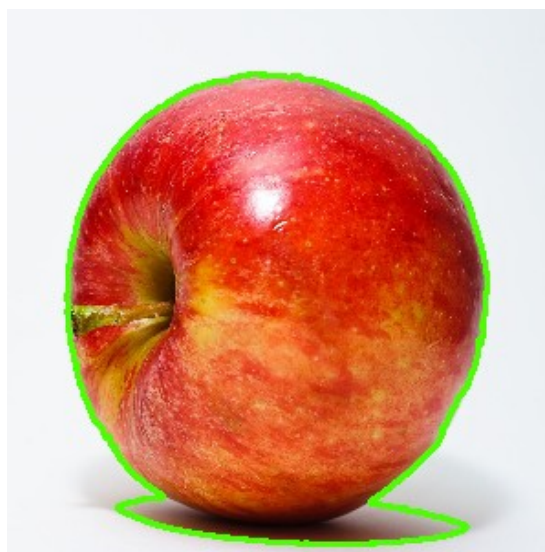


Рисунок 3 - Изображение с контурами

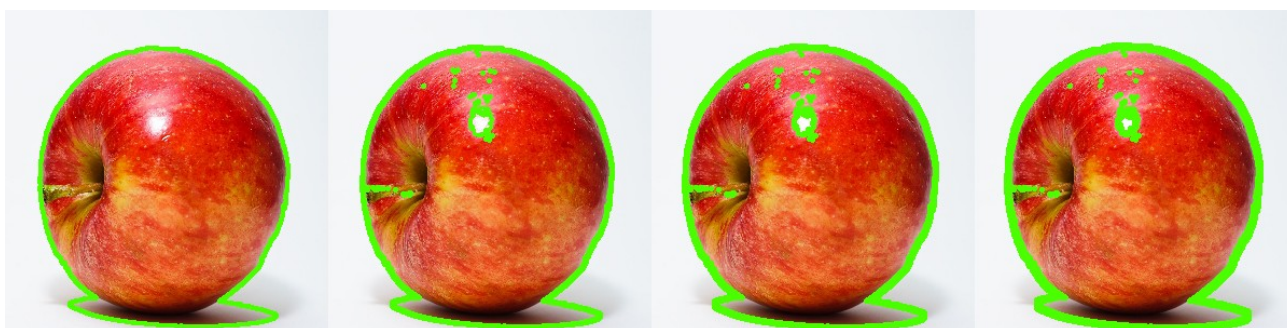


Рисунок 4 - Результат задания 6 (различные параметры mode)

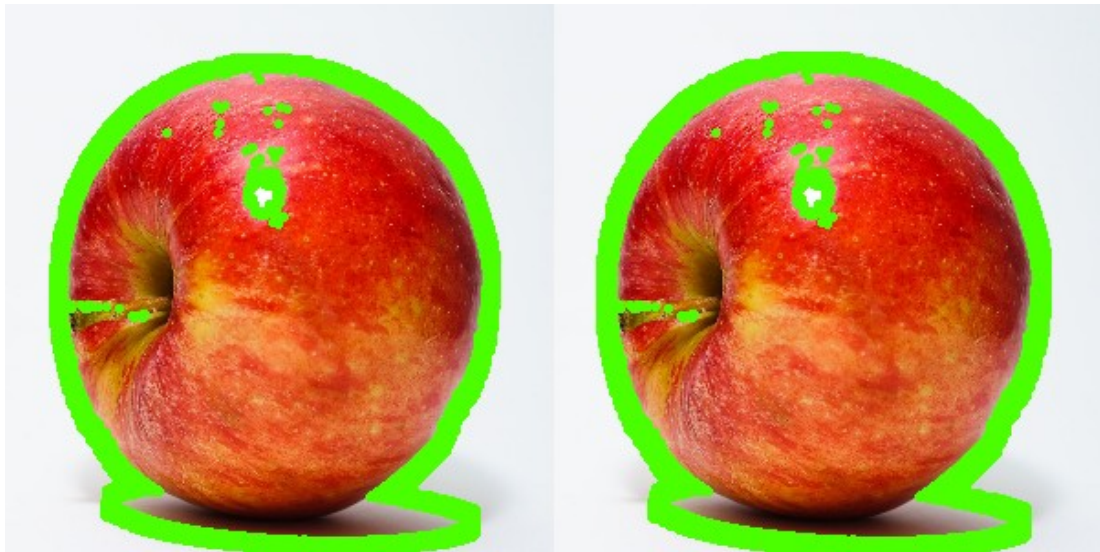


Рисунок 5 - Результат задания 7 (различные параметры method)

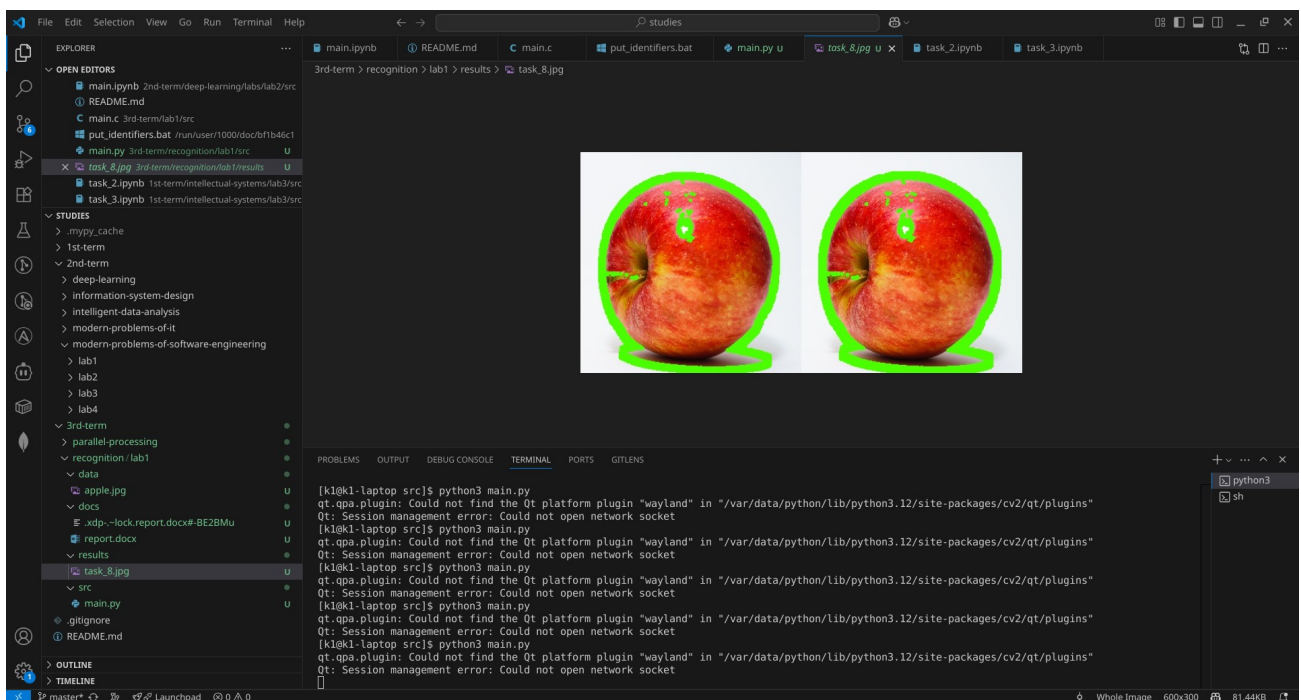


Рисунок 6 - Результат задания 8

Вывод: в процессе выполнения лабораторной работы рассмотрены технические этапы подготовки инфраструктуры и продемонстрировано использование базовых функций библиотеки OpenCV на простых практических примерах

Листинг программы

```
# -*- coding: utf-8 -*-
"""
    Программные системы распознавания и обработки информации.
    Лабораторная работа 1.
    Сборка и установка библиотеки opencv.
"""
import cv2
import typing as t

IMAGE_PATH = "../data/apple.jpg"
RESULT_IMAGE_PATH = "../results/task_8.jpg"
RESIZE_WIDTH = 300
RESIZE_HEIGHT = 300

def load_image(path: str) -> cv2.typing.MatLike:
    """Загрузить изображение из OpenCV."""
    image = cv2.imread(path)

    if image is None:
        raise ValueError('Error occupied when loading image')

    return cv2.resize(image, (RESIZE_WIDTH, RESIZE_HEIGHT))

def display_image(title: str, image: cv2.typing.MatLike):
    """Вывести изображение из OpenCV."""
    cv2.imshow(title, image)
    cv2.waitKey()

def save_image(path: str, image: cv2.typing.MatLike):
    """Сохранить изображение по заданному пути."""
    cv2.imwrite(path, image)

def rgb_to_gray(image: cv2.typing.MatLike) -> cv2.typing.MatLike:
    """Преобразовать изображение из RGB в градации серого."""
    gray_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    return gray_image

def binarize_image( grayscale_image: cv2.typing.MatLike) ->
cv2.typing.MatLike:
    """Бинаризовать изображение."""
    _, thresh = cv2.threshold(grayscale_image, 190, 255,
cv2.THRESH_BINARY_INV)
    return thresh
```

```

def find_contours(
    grayscale_image: cv2.typing.MatLike,
    mode: int = cv2.RETR_EXTERNAL,
    method: int = cv2.CHAIN_APPROX_SIMPLE,
) -> t.Sequence[cv2.typing.MatLike]:
    """Найти контуры изображения."""
    im = binarize_image(grayscale_image)
    contours, _ = cv2.findContours(im, mode, method)
    return contours

def draw_contours(
    image: cv2.typing.MatLike,
    contours: t.Sequence[cv2.typing.MatLike],
) -> cv2.typing.MatLike:
    """Нарисовать контуры на изображении."""
    res_image = cv2.drawContours(image, contours, -1, (0, 255, 75), 2)
    return res_image

def task_5(image: cv2.typing.MatLike):
    """
    Отобразить в одном окне изображение в оттенках серого и
    бинаризованное.
    """
    display_image("Original", image)

    grayscale_image = rgb_to_gray(image)
    h_concat = cv2.hconcat([
        grayscale_image,
        binarize_image(grayscale_image),
    ])
    display_image("Grayscale and Binarized Image", h_concat)

    contours = find_contours(grayscale_image)
    contoured_image = draw_contours(image, contours)
    display_image("Contoured Image", contoured_image)

def task_6(image: cv2.typing.MatLike):
    """
    Отобразить все изображения из задачи 3 в одном окне.
    Эксперименты со всеми возможными значениями параметра
    mode в функции findContours.
    """
    res = list[cv2.typing.MatLike]()
    for i in range(cv2.RETR_EXTERNAL, cv2.RETR_FLOODFILL):
        current = draw_contours(
            image=image,
            contours=find_contours(rgb_to_gray(image), mode=i),

```



```

        )
        res.append(cv2.resize(current, (RESIZE_WIDTH, RESIZE_HEIGHT)))

    h_concat = cv2.hconcat(res)
    display_image("Task 6 Results", h_concat)

def task_7(image: cv2.typing.MatLike):
    """
    Отобразить все изображения из задачи 4 в одном окне.
    Эксперименты со всеми возможными значениями параметра
    method в функции findContours.
    """
    res = []
    for i in range(cv2.CHAIN_APPROX_SIMPLE,
cv2.CHAIN_APPROX_TC89_KCOS, 1):
        img = draw_contours(
            image=image,
            contours=find_contours(
                grayscale_image=rgb_to_gray(image),
                method=i,
            ),
        )
        res.append(cv2.resize(img, (RESIZE_WIDTH, RESIZE_HEIGHT)))

    h_concat = cv2.hconcat(res)
    display_image("Task 7 Results", h_concat)
    return h_concat

def task_8(image: cv2.typing.MatLike):
    """
    Сохранить результаты задачи 2 в файл.
    """
    result = task_7(image)
    save_image(RESULT_IMAGE_PATH, result)

if __name__ == '__main__':
    image = load_image(IMAGE_PATH)
    task_5(image)
    task_6(image)
    task_7(image)
    task_8(image)

```