

Министерство науки и высшего образования Российской Федерации  
Муромский институт (филиал)  
Федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет \_\_\_\_\_ ИТР \_\_\_\_\_

Кафедра \_\_\_\_\_ ПИН \_\_\_\_\_

## ЛАБОРАТОРНАЯ РАБОТА №4

По \_\_\_\_\_ Цифровая обработка информации \_\_\_\_\_

Тема СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЯ И ВЫДЕЛЕНИЕ  
КОНТУРОВ \_\_\_\_\_

Руководитель

Белякова А.С.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Студент \_\_\_\_\_ ПИН - 121 \_\_\_\_\_

\_\_\_\_\_  
(группа)

Ермилов М.В.

\_\_\_\_\_  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Муром 2024

# Лабораторная работа №4

Тема: СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЯ И ВЫДЕЛЕНИЕ КОНТУРОВ

Цель работы – изучение операций по сегментации изображений, функций, реализующих операции по сегментации изображения, и приобретение практических навыков использования этих функций.

## Ход работы:

### 1. Исходный код Python:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Загрузка изображения
image_path = 'IDRiD_10.jpg' # Путь к изображению
image = cv2.imread(image_path)
image_gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# 1. Метод выращивания областей
def region_growing(image, seed, threshold=20):
    output_image = np.zeros_like(image)
    visited = np.zeros_like(image, dtype=bool)
    region = [seed]
    visited[seed] = True
    while region:
        x, y = region.pop()
        # Проходим по соседним пикселям
        for dx in [-1, 1]:
            for dy in [-1, 1]:
                nx, ny = x + dx, y + dy
                if 0 <= nx < image.shape[0] and 0 <= ny < image.shape[1] and not visited[nx, ny]:
                    if abs(int(image[nx, ny]) - int(image[x, y])) < threshold:
                        output_image[nx, ny] = 255
                        visited[nx, ny] = True
                        region.append((nx, ny))
    return output_image

# 2. Метод разделения
def segmentation_by_threshold(image, low_thresh=100, high_thresh=200):
    _, segmented_image = cv2.threshold(image, low_thresh, high_thresh, cv2.THRESH_BINARY)
    return segmented_image

# 3. Сегментация по гистограмме
def segmentation_by_histogram(image, threshold=127):
    hist, bins = np.histogram(image.flatten(), 256, [0, 256])
    cumulative_sum = np.cumsum(hist)
    total_pixels = cumulative_sum[-1]
    threshold = np.argmax(cumulative_sum > total_pixels / 2)
    _, segmented_image = cv2.threshold(image, threshold, 255, cv2.THRESH_BINARY)
    return segmented_image
```

|           |      |               |         |      |                                                    |      |        |
|-----------|------|---------------|---------|------|----------------------------------------------------|------|--------|
|           |      |               |         |      | МИВУ 09.03.04 - 10.004                             |      |        |
| Изм.      | Лист | № докум.      | Подпись | Дата | СЕГМЕНТАЦИЯ<br>ИЗОБРАЖЕНИЯ И ВЫДЕЛЕНИЕ<br>КОНТУРОВ |      |        |
| Разраб.   |      | Ермилов М.В.  |         |      |                                                    |      |        |
| Провер.   |      | Белякова А.С. |         |      |                                                    |      |        |
| Реценз.   |      |               |         |      |                                                    |      |        |
| Н. Контр. |      |               |         |      |                                                    |      |        |
| Утверд.   |      |               |         |      | МИ ВлГУ ПИН-121                                    |      |        |
|           |      |               |         |      | Лит.                                               | Лист | Листов |
|           |      |               |         |      |                                                    | 2    | 4      |

```

# 4. Яркой срез
def brightness_slicing(image, low=50, high=150):
    sliced_image = np.zeros_like(image)
    sliced_image[(image >= low) & (image <= high)] = 255
    return sliced_image

# Применение методов
seed_point = (50, 50) # Начальная точка для выращивания области

region_grown_image = region_growing(image_gray, seed_point)
segmented_by_threshold_image = segmentation_by_threshold(image_gray)
segmented_by_histogram_image = segmentation_by_histogram(image_gray)
brightness_sliced_image = brightness_slicing(image_gray)

# Отображение результатов
plt.figure(figsize=(10, 8))

plt.subplot(2, 2, 1)
plt.imshow(region_grown_image, cmap='gray')
plt.title('Метод выращивания областей')
plt.axis('off')

plt.subplot(2, 2, 2)
plt.imshow(segmented_by_threshold_image, cmap='gray')
plt.title('Метод разделения')
plt.axis('off')

plt.subplot(2, 2, 3)
plt.imshow(segmented_by_histogram_image, cmap='gray')
plt.title('Сегментация по гистограмме')
plt.axis('off')

plt.subplot(2, 2, 4)
plt.imshow(brightness_sliced_image, cmap='gray')
plt.title('Яркой срез')
plt.axis('off')

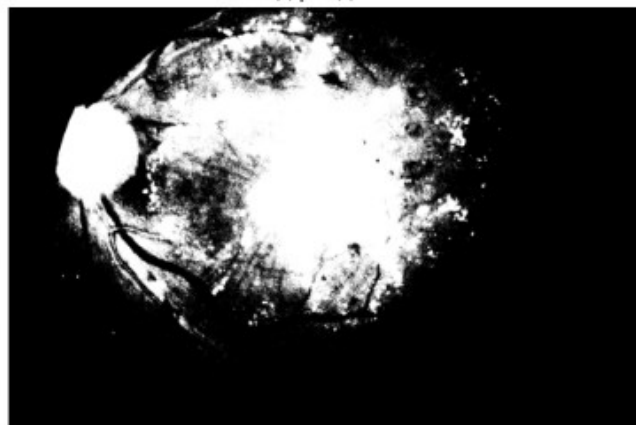
plt.tight_layout()
plt.show()

```

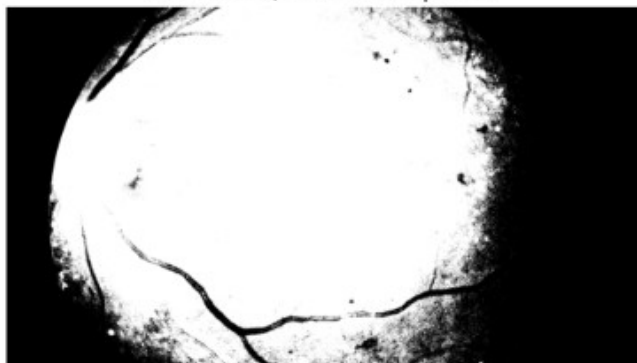
Метод выращивания областей



Метод разделения



Сегментация по гистограмме



Яркостной срез



Рисунок 1 – результат обработки фото

Вывод: в ходе лабораторной работы было изучено операции по сегментации изображений, функций, реализующих операции по сегментации изображения, и приобретение практических навыков использования этих функций