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

# 4. Яркостной 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('Яркостной срез')

plt.axis('off')

plt.tight\_layout()

plt.show()

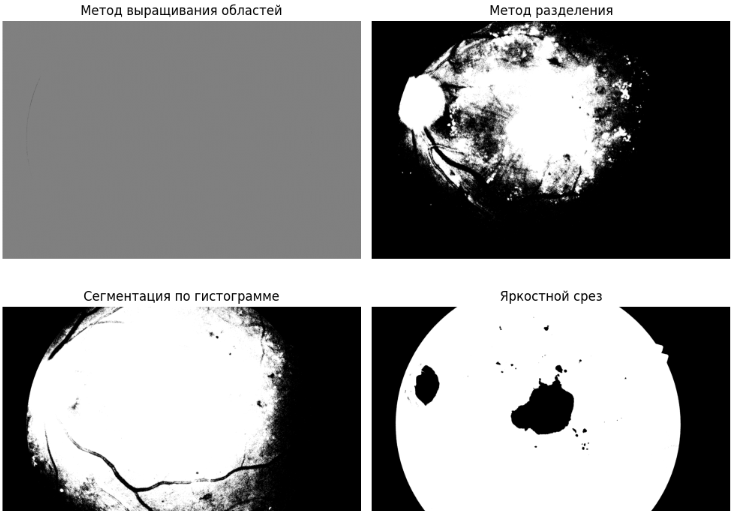


Рисунок 1 – результат обработки фото

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