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 №5*

По Цифровая обработка информации

Тема РАСПОЗНАВАНИЕ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ

Руководитель

Белякова А.С.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИН - 121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

**Лабораторная работа №5**

Цель работы: изучить и освоить алгоритмы вычисления логических и геометрических признаков бинарных и полутоновых изображений. Выделение контура кровеносных сосудов.

**Ход работы:**

1. Исходный код Python:

import numpy as np

import cv2

import matplotlib.pyplot as plt

# Функция для создания круглой маски

def create\_circular\_mask(image\_shape):

height, width = image\_shape[:2]

center = (width // 2, height // 2)

radius = min(center[0], center[1]) - 10 # Радиус круга с небольшим отступом от краев

mask = np.zeros((height, width), dtype=np.uint8)

cv2.circle(mask, center, radius, 255, -1) # Заполняем круг белым цветом

return mask

# Функция для применения маски к изображению

def apply\_mask(image, mask):

return cv2.bitwise\_and(image, image, mask=mask)

# Функция для подсчета длины и количества толстых капилляров (предположим, что она определена)

def count\_thick\_capillaries(image, min\_perimeter):

# Пример реализации функции

contours, \_ = cv2.findContours(image, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

thick\_capillaries = [cnt for cnt in contours if cv2.arcLength(cnt, True) > min\_perimeter]

capillary\_length = sum(cv2.arcLength(cnt, True) for cnt in thick\_capillaries)

capillary\_count = len(thick\_capillaries)

return capillary\_length, capillary\_count, thick\_capillaries

# Загрузка и предварительная обработка изображения (предположим, что изображение уже загружено)

image = cv2.imread('IDRiD\_10.jpg')

image\_gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

# Создаем круглую маску

circular\_mask = create\_circular\_mask(image\_gray.shape)

# Применяем маску к сегментированному изображению

segmented\_with\_mask = apply\_mask(segmented\_by\_histogram\_image, circular\_mask)

# Подсчет длины и количества толстых капилляров на изображении с маской

min\_perimeter = 50 # Пример значения минимального периметра

capillary\_length, capillary\_count, thick\_capillaries = count\_thick\_capillaries(segmented\_with\_mask, min\_perimeter)

# Создаем изображение для отображения выделенных капилляров с учетом маски

image\_with\_capillaries\_masked = image.copy()

cv2.drawContours(image\_with\_capillaries\_masked, thick\_capillaries, -1, (0, 255, 0), 2)

# Отображение результатов

plt.figure(figsize=(10, 8))

# Изображение с выделенными капиллярами и маской

plt.subplot(1, 2, 1)

plt.imshow(cv2.cvtColor(image\_with\_capillaries\_masked, cv2.COLOR\_BGR2RGB))

plt.title('Толстые капилляры с маской')

plt.axis('off')

# Сегментированное изображение

plt.subplot(1, 2, 2)

plt.imshow(segmented\_with\_mask, cmap='gray')

plt.title('Сегментация с маской')

plt.axis('off')

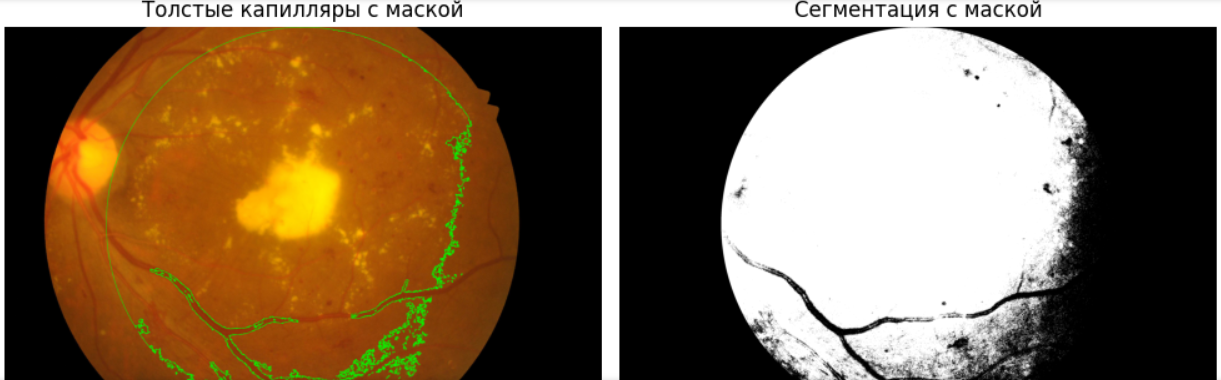
plt.tight\_layout()

plt.show()

# Вывод в консоль

print(f'Общая длина толстых капилляров: {capillary\_length:.2f} пикселей')

print(f'Количество толстых капилляров: {capillary\_count}')



Общая длина толстых капилляров: 81045.94 пикселей

Количество толстых капилляров: 242

Рисунок 1 – результат обработки фото

Вывод: в ходе лабораторной работы было изучены различные признаки объектов на изображении, используемые для распознавания кровеносных сосудов.