Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 4

по курсу «Цифровая обработка изображение»

Выполнил:  
студент группы 22ВВИм1:

Милованов А.С.

Приняли:

Митрохин М.А  
Панков А.А.

Пенза 2023

# Цель работы: ознакомиться с библиотекой OpenCV и получить навык её использования. Применить теоретические знания об улучшении изображения

# Порядок выполнения работы:

# Задание:

# Преобразовать/улучшить исходное изображение

# D:\Anton\Универ_Мага\2_Курс\1_семестр\Лабы\Обработка_Изображения_Митрохин\image-processing\Лабораторная 4\input.jpg

Рисунок Исходное изображение

# Листинг программы:

#!/usr/bin/env python

# coding: utf-8

# In[1]:

import numpy as np

import cv2 as cv

import matplotlib.pyplot as plt

# In[2]:

input\_image = cv.imread('input.jpg')

# In[3]:

plt.imshow(input\_image)

# In[4]:

image = input\_image

gray\_image = cv.cvtColor(input\_image, cv.COLOR\_BGR2GRAY)

channels = [0]

histSize = [256]

range = [0, 256]

gs = plt.GridSpec(1, 2)

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

plt.subplot(gs[0])

plt.imshow(gray\_image, cmap='gray')

plt.xticks([]), plt.yticks([])

plt.subplot(gs[1])

plt.hist(gray\_image.reshape(-1), 256, range)

plt.show()

# In[5]:

threshold = 115

image = gray\_image

ret, binary\_image = cv.threshold(image, threshold, 255, cv.THRESH\_BINARY)

plt.title("Binary")

plt.imshow(binary\_image,"gray")

# In[6]:

cv.imwrite("binary\_image.jpg", binary\_image)

# In[7]:

image = binary\_image

kernel = np.ones((4, 4), np.uint8)

dilation = cv.dilate(image, kernel, iterations=1)

erosion = cv.erode(image, kernel, iterations=1)

opening = cv.morphologyEx(image, cv.MORPH\_OPEN, kernel)

closing = cv.morphologyEx(image, cv.MORPH\_CLOSE, kernel)

closeAndOpen = cv.morphologyEx(opening, cv.MORPH\_CLOSE, kernel)

# вывод

plt.figure(figsize=(15, 8))

gs = plt.GridSpec(2, 3)

titles = ['Зашумленное изображение', 'Дилатация', 'Эрозия', 'Открытие', 'Закрытие',

'Последовательное закрытие и открытие']

outImages = [image, dilation, erosion, opening, closing, closeAndOpen]

for i in np.arange(len(outImages)):

plt.subplot(gs[i])

plt.xticks([]), plt.yticks([])

plt.title(titles[i])

plt.imshow(outImages[i], cmap='gray')

plt.show()

# In[8]:

cv.imwrite("erosion.jpg", erosion)

# In[9]:

image = erosion

kernel = np.ones((3, 3), np.uint8)

dilation = cv.dilate(image, kernel, iterations=1)

erosion = cv.erode(image, kernel, iterations=1)

opening = cv.morphologyEx(image, cv.MORPH\_OPEN, kernel)

closing = cv.morphologyEx(image, cv.MORPH\_CLOSE, kernel)

closeAndOpen = cv.morphologyEx(opening, cv.MORPH\_CLOSE, kernel)

# вывод

plt.figure(figsize=(15, 8))

gs = plt.GridSpec(2, 3)

titles = ['Зашумленное изображение', 'Дилатация', 'Эрозия', 'Открытие', 'Закрытие',

'Последовательное закрытие и открытие']

outImages = [image, dilation, erosion, opening, closing, closeAndOpen]

for i in np.arange(len(outImages)):

plt.subplot(gs[i])

plt.xticks([]), plt.yticks([])

plt.title(titles[i])

plt.imshow(outImages[i], cmap='gray')

plt.show()

# In[10]:

plt.imshow(closeAndOpen,"gray")

# In[11]:

cv.imwrite("closeAndOpen.jpg", closeAndOpen)

# In[ ]:

# Результат работы программы:

# 

Рисунок Гистограмма исходного изображения

# 

Рисунок Результат применения бинаризации к исходному изображению

# 

Рисунок Результат применения математической морфологии Серры к бинаризированному изображению

# 

Рисунок Результат применения математической морфологии Серры к изображению с Эрозией изображению

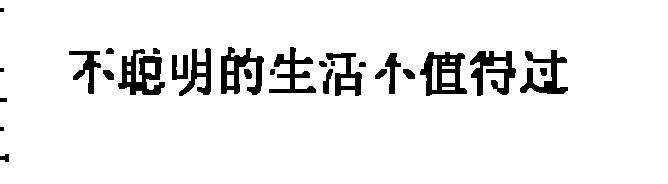


Рисунок Результат последовательного закрытия и открытия

# 

Рисунок Перевод текста с изображения

# Вывод:

# В ходе выполнения лабораторной работы были получены навыки использования методов бинаризации

# 