Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе № 6

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 методах сегментации изображений.

# Порядок выполнения работы:

# Задание:

# Выделить желтые шарыD:\Anton\Универ_Мага\2_Курс\1_семестр\Лабы\Обработка_Изображения_Митрохин\image-processing\Лабораторная 5\input.jpg

Рисунок Исходное изображение

# Листинг программы:

import sys

import numpy as np

import cv2 as cv

from sklearn.cluster import MeanShift, estimate\_bandwidth

from sklearn.cluster import KMeans

from skimage.feature import peak\_local\_max

from skimage.segmentation import watershed

from skimage import data

from scipy import ndimage

import matplotlib.pyplot as plt

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

image = input\_image

image\_rgb = cv.cvtColor(input\_image, cv.COLOR\_BGR2RGB)

plt.imshow(image\_rgb)

image\_hsv = cv.cvtColor(image\_rgb, cv.COLOR\_RGB2HSV)

plt.imshow(image\_hsv)

import sys

import numpy as np

import cv2 as cv

from sklearn.cluster import MeanShift, estimate\_bandwidth

from sklearn.cluster import KMeans

from skimage.feature import peak\_local\_max

from skimage.segmentation import watershed

from skimage import data

from scipy import ndimage

import matplotlib.pyplot as plt

gray =cv.cvtColor(image\_hsv, cv.COLOR\_RGB2GRAY)

## Методы кластеризации. K-средних

# Преобразуем изображение в оттенках серого в одномерный массив

pixels = gray.reshape(-1, 1)

# Задаем число кластеров для сегментации

K = 2

# С помощью библиотеки sklearn.cluster import KMeans проводим кластеризацию по яркости

kmeans = KMeans(n\_clusters=K, random\_state=0)

labels = kmeans.fit\_predict(pixels)

cluster\_centers = kmeans.cluster\_centers\_

print (np.uint8(cluster\_centers))

# Каждому пикселю назначаем значение из центра кластера

segments = np.uint8(cluster\_centers[labels].reshape(gray.shape))

print(segments)

plt.imshow(segments, cmap='Set3')

segments[segments==109 ] = 0

result = cv.bitwise\_and(image\_hsv, image\_hsv, mask=segments)

# Отобразим избражения

plt.figure(figsize=(15,20))

plt.subplot(1, 3, 1)

plt.imshow(cv.cvtColor(gray, cv.COLOR\_GRAY2RGB))

plt.subplot(1, 3, 3)

plt.imshow(result)

plt.show()

gray =cv.cvtColor(rgb\_result, cv.COLOR\_RGB2GRAY)

## Методы кластеризации. K-средних

# Преобразуем изображение в оттенках серого в одномерный массив

pixels = gray.reshape(-1, 1)

# Задаем число кластеров для сегментации

K = 3

# С помощью библиотеки sklearn.cluster import KMeans проводим кластеризацию по яркости

kmeans = KMeans(n\_clusters=K, random\_state=0)

labels = kmeans.fit\_predict(pixels)

cluster\_centers = kmeans.cluster\_centers\_

print (np.uint8(cluster\_centers))

# Каждому пикселю назначаем значение из центра кластера

segments = np.uint8(cluster\_centers[labels].reshape(gray.shape))

print(segments)

plt.imshow(segments, cmap='Set3')

segments[segments!=182 ] = 0

resultt = cv.bitwise\_and(image\_hsv, image\_hsv, mask=segments)

# Отобразим избражения

plt.figure(figsize=(15,20))

plt.subplot(1, 3, 1)

plt.imshow(cv.cvtColor(gray, cv.COLOR\_GRAY2RGB))

plt.subplot(1, 3, 3)

plt.imshow(resultt)

plt.show()

# Результат работы программы:

# 

Рисунок 2 Кластеризация

# 

Рисунок 3 Применение маски

# 

Рисунок 4 Повторная кластеризация

# 

Рисунок 5 Результат

# Вывод:

# В ходе выполнения лабораторной работы были получены навыки применения методов сегментации изображения

# 