

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

Отчёт по практическому занятию №3.9

**«Бинарные изображения, основные характеристики
бинарных изображений»**

по дисциплине «Теории распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1
Образцова М.Д. « » _____ 20__ г.
Подпись студента _____
Работа защищена « » _____ 20__ г.
Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

Цель работы: изучение методов цифровой обработки бинарных изображений, геометрических характеристик этих изображений, способов получения дополнительных параметров бинарных изображений. Изучение основных функций OpenCv, применяемых для цифровой обработки бинарных изображений.

Выполнение работы

```
In [8]: M = cv2.moments(cnt)
print(M)

{'m00': 3682561.0, 'm10': 3533417279.5, 'm01': 3533417279.5, 'm20': 4520418506240.333, 'm11': 3390313879680.25, 'm02': 4520418506240.333, 'm30': 6506012335106149.0, 'm21': 4337341556737600.0, 'm12': 4337341556737600.0, 'm03': 6506012335106400.0, 'mu20': 1130104626560.0825, 'mu11': -0.00048828125, 'mu02': 1130104626560.0825, 'mu30': -250.0, 'mu21': 0.75, 'mu12': 0.75, 'mu03': 1.0, 'nu20': 0.0833333333333329, 'nu11': -3.6005607985627815e-17, 'nu02': 0.0833333333333329, 'nu30': -9.60649884765057e-15, 'nu21': 2.881949654295171e-17, 'nu12': 2.881949654295171e-17, 'nu03': 3.842599539060228e-17}
```

```
In [9]: x, y, w, h = cv2.boundingRect(cnt)
print(x, y, w, h)

0 0 1920 1920
```

```
In [59]: imag = cv2.rectangle(imag, (x, y), (x+w, y+h), (0, 255, 0), 2)
cv2.imshow('Rectan', imag)
```

```
In [ ]: asprat = float(w) / h
rectar = w * h
extent = float(ar) / rectar
eqdiam = np.sqrt(4*ar / np.pi)

print(asprat, extent)
print(eqdiam)
cv2.waitKey(0)

1.0 0.9989586046086944
2165.3596216562887
```

3.2.1. Маска и пиксельные точки

```
In [29]: import cv2
import numpy as np

In [30]: img = cv2.imread("cat.jpeg", 0)
mask = np.zeros(img.shape, np.uint8)

cv2.drawContours(mask, [cnt], 0, 255, -1)
pixpoints = np.transpose(np.nonzero(mask))
pixpoints = cv2.findNonZero(mask)
```

3.2.2. Максимальное и минимальное значения и их координаты

```
In [38]: minval, maxval, minloc, maxloc = cv2.minMaxLoc(img, mask=mask)
print(minval, maxval, minloc, maxloc)

0.0 255.0 (877, 0) (695, 402)
```

3.2.3. Крайние точки

```
In [37]: leftmost = tuple(cnt[cnt[:, :, 0].argmin()][0])
rightmost = tuple(cnt[cnt[:, :, 0].argmax()][0])
topmost = tuple(cnt[cnt[:, :, 1].argmin()][0])
bottommost = tuple(cnt[cnt[:, :, 1].argmax()][0])
print(leftmost, rightmost, topmost, bottommost)

(0, 0) (1919, 0) (0, 0) (1919, 1919)
```

Самостоятельное задание ¶

Задача по нахождению границ объекта на бинарном изображении с помощью алгоритма Кэнни

Алгоритм Кэнни - это классический алгоритм обнаружения границ на изображении, состоит из нескольких шагов

-Сглаживание (фильтрация) изображения с помощью гауссового фильтра.

-Определение градиента яркости на изображении.

-Подавление не-максимумов.

-Пороговая обработка.

-Соединение границ.

```
In [1]: import cv2
import numpy as np
from matplotlib import pyplot as plt
```

Загрузим изображение и преобразуем его в черно-белое бинарное изображение

```
In [6]: img = cv2.imread("cats.jpg")
img_binary = cv2.imread("cats.jpg", cv2.IMREAD_GRAYSCALE)
```

Преобразуем его в черно-белое бинарное изображение

Определим пороговое значение, 128 - это середина черного и белого в шкале

```
In [7]: thresh = 128

img_binary = cv2.threshold(img_binary, thresh, 255, cv2.THRESH_BINARY)[1]

cv2.imwrite('b_w.png', img_binary)
```

Out[7]: True

Применим алгоритм Кэнни для нахождения границ объекта

```
In [8]: img_binary = cv2.Canny(img_binary, 100, 200)
```

Отобразим результат

```
In [9]: plt.subplot(121), plt.imshow(img)
plt.title('Original Image')
plt.subplot(122), plt.imshow(img_binary)
plt.title('Canny Image')
plt.show()
```

