

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ**

ФЕДЕРАЦИИ

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

«Основы цифровой обработки изображений в OpenCv»

Отчет по лабораторной работе № 7 (1)

по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1

Кучеренко С. Ю. « » 2023г.

Подпись студента _____

Работа защищена « » _____ 2023г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2023

Цель работы: изучение типов изображений, способов их формирования. Изучение основных функций OpenCv, применяемых для цифровой обработки изображений.

1. Считывание изображения и вывод его на экран, запись изображения в файл.
2. Вывод свойств изображения и сформированной матрицы на экран.
3. Доступ к изображению для изменения значений цвета пикселей.
4. Создание бинарного изображения и его негатива.
5. Применение библиотеки matplotlib для вывода нескольких изображений в общем окне.
6. Выделение и взятие в рамку определенного региона изображения.
7. Уменьшение размера изображения и вывод матрицы на экран.
8. Знакомство с процессом дискретизации и квантования изображения.
9. Приобретение практических навыков использования этих функций.

Выполнение работы:

Задание 1.1.

Считать файл полноцветного изображения cat.jpg, создать для него матрицу изображения, затем вывести сначала полутоновое, затем цветное изображение на экран. Перед выполнением задания получить согласно номеру в списке группы свой файл с изображением.

```
In [2]: import cv2
```

```
In [ ]: img = cv2.imread('cat.jpg', 0)
cv2.imshow('image_1', img)
cv2.waitKey(0)
```

```
In [ ]: img = cv2.imread('cat.jpg', cv2.IMREAD_GRAYSCALE)
cv2.imshow('image_2', img)
cv2.waitKey(0)
```

Задание 1.2.

Используя код задания 1.1, в функции cv2.imread(,) присвоить флагу значение 1, затем вывести изображение на экран. Выполнить этот же код, заменив в функции cv2.imread('cat.jpg', 1) флаг 1 на флаг cv2.IMREAD_COLOR.

```
In [ ]: img = cv2.imread('cat.jpg', 1)
cv2.imshow('image_3', img)
cv2.waitKey(0)
```

```
In [ ]: img = cv2.imread('cat.jpg', cv2.IMREAD_COLOR)
cv2.imshow('image_4', img)
cv2.waitKey(0)
```

Задание 1.3.

Сформировать матрицу изображения, записать ее в файл с расширением png. Изображение, записанное в этом файле, вывести на экран.

```
In [ ]: img = cv2.imread('cat.jpg')
# запись изображения из матрицы в файл
cv2.imwrite('img.png', img)
img = cv2.imread('img.png')
cv2.imshow('image_5', img)
cv2.waitKey(0)
```

Задание 1.4.

Сформировать матрицу, у которой выше диагонали единицы, а ниже – нули, записать ее в файл, затем считать файл и вывести на экран. Строим массив 28*28:

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

n = 28
a = np.ones([28, 28])
for i in range(n):
    a[i][i] = 1

for i in range(n):
    for j in range(0, i):
        a[i][j] = 0

cv2.imwrite('ris.png', a)
img = cv2.imread('ris.png', 0)
cv2.imshow('image_6', img)
print(img)
cv2.waitKey(0)
```

```
[[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]]
```

Задание 1.5.

Вывести свойства матрицы изображения на экран.

```
In [1]: import cv2
import numpy as np

img = cv2.imread('cat.jpg', 0)
cv2.imshow('image', img)

print(type(img))
print(img.shape)
print(img.size)
print(img.dtype)
```

```
<class 'numpy.ndarray'>
(457, 685)
313045
uint8
```

Задание 1.6.

Определить с помощью функции `print(img.shape)` максимальное число пикселей по ширине и высоте изображения. Выбрать координаты так, чтобы они не выходили за пределы размеров изображения. Задать координату по горизонтали равной сумме номера по списку группы плюс 70, по вертикали равной сумме номера по списку группы плюс 50.

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

img = cv2.imread('cat.jpg')
print(img.shape)

cv2.waitKey(0)
px = img[100, 150]
print(px)
cv2.waitKey(0)

blue = img[100, 150, 0]
print(blue)

img[100, 150] = [105, 139, 185]
print(img[100, 150])

img.item(100,150,2)
img.itemset((100,150,2),100)

(457, 685, 3)
```

Задание 1.7.

Считать файл полноцветного изображения cat.jpg, создать для него два места в окне в ширину и два места в высоту. Преобразовать матрицу цветного изображения в полутоновое, из него, используя функцию cv2.threshold, получить бинарное монохромное изображение. Из бинарного монохромного изображения получить его негатив.

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

img = cv2.imread("cat.jpg")
imag = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

plt.subplot(221)
plt.imshow(imag)
plt.axis('off')

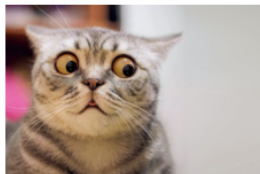
gray_img = cv2.imread("cat.jpg", 0)
im_bw = cv2.threshold(gray_img, 128, 255, cv2.THRESH_BINARY)[1]

plt.subplot(222)
plt.imshow(im_bw, 'gray')
plt.axis("off")

im_bwa = cv2.threshold(imag, 128, 255, cv2.THRESH_BINARY)[1]

plt.subplot(223)
plt.imshow(im_bwa)
plt.axis("off")

im_bwb = cv2.threshold(gray_img, 128, 255, cv2.THRESH_BINARY_INV)[1]
plt.subplot(224)
plt.imshow(im_bwb, 'gray')
plt.axis("off")
plt.show()
cv2.waitKey(0)
```



Задание 1.8.

На заданном изображении выделить его характерный участок.

```
In [1]: import cv2

img = cv2.imread("avto.jpg")
image = cv2.rectangle(img, (280, 340), (330, 390), (0, 0, 255), 2)
cv2.imshow('selected', img)
```

Задание 1.9.

Уменьшить заданное изображение и вывести на печать матрицу уменьшенного изображения. Нам надо сохранить соотношение сторон, чтобы изображение не исказилось при уменьшении.

In [1]: `import cv2`

```
img = cv2.imread('avto.jpg')
final_wide = 200
r = float(final_wide) / img.shape[1]
dim = (final_wide, int(img.shape[0]*r))
resized = cv2.resize(img,dim,interpolation=cv2.INTER_AREA)
cv2.imshow("Resize image", resized)
print(resized.shape)
print(resized)
img = cv2.imread('avto.jpg', 0)
cv2.imshow('car', img)
print(img)
```

```
(112, 200, 3)
[[ [ 8  8  8]
  [ 8  8  8]
  [ 8  8  8]
  ...
  [ 11 10 14]
  [ 12 11 15]
  [ 12 11 15]]

[[ [ 8  8  8]
  [ 8  8  8]
  [ 8  8  8]
  ...
  [ 11 10 14]
  [ 12 11 15]
  [ 12 11 15]]

[[ [ 8  8  8]
  [ 8  8  8]
  [ 8  8  8]
  ...
  [ 11 10 14]
  [ 12 11 15]
  [ 12 11 15]]

...

[[ [ 12 12 12]
  [ 10 10 10]
  [ 90 90 90]
  ...
  [ 19 17 22]
  [ 18 17 21]
  [ 20 19 23]]

[[ [ 15 15 15]
  [125 125 125]
  [141 141 141]
  ...
  [ 22 19 23]
  [ 19 18 22]
  [ 20 19 23]]

[[[146 146 146]
  [116 116 116]
  [ 12 12 12]
  ...
  [ 22 19 23]
  [ 20 19 23]
```

Задание 1.10.

Считать цветное изображение, конвертировать его в полутоновое, затем получить негатив полутонового изображения.

```
In [ ]: import cv2

img = cv2.imread("avto.jpg", 1)
img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
img = cv2.bitwise_not(img)
cv2.imshow('Img',img)
cv2.waitKey(0)
```