

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Основы цифровой обработки изображений в OpenCv»

ОТЧЕТ
по лабораторной работе №7
дисциплины
«Технологии распознавания образов»

Выполнил:
Мизин Глеб Егорович
2 курс, группа ПИЖ-б-о-21-1,
011.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

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.

Импортируем модуль cv2:

```
In 3 1 import cv2

In 4 1 image = cv2.imread('Cat.jpg', 0)
    2 cv2.imshow('Cat image', image)
    3 cv2.waitKey(0)
    4 cv2.destroyAllWindows()

In _ 1 image = cv2.imread('Cat.jpg', cv2.IMREAD_GRAYSCALE)
    2 cv2.imshow('Cat image', image)
    3 cv2.waitKey(0)
    4 cv2.destroyAllWindows()
```

Рисунок 1 – Задание 1.1

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 3 1 import cv2

In 4 1 image = cv2.imread('Cat.jpg', 1)
    2 cv2.imshow('Cat image', image)
    3 cv2.waitKey(0)
    4 cv2.destroyAllWindows()

In 5 1 image = cv2.imread('Cat.jpg', cv2.IMREAD_COLOR)
    2 cv2.imshow('Cat image', image)
    3 cv2.waitKey(0)
    4 cv2.destroyAllWindows()
```

Рисунок 2 – Задание 1.2

Задание 1.3

Запись изображения в файл

```
In 1 1 import cv2

In 2 1 img = cv2.imread('cat.jpg')
      2 cv2.imwrite('img.png', img)
      3 img = cv2.imread('img.png')
      4 cv2.imshow('image', img)
      5 cv2.waitKey(0)
      6 cv2.destroyAllWindows()
```

Рисунок 3 – Задание 1.3

Задание 1.4

Задание 1.4. Сформировать матрицу, у которой выше диагонали единицы, а ниже – нули, записать ее в файл, затем считать файл и вывести на экран.

Импортируем необходимые библиотеки

```
In 1 1 import ...
```

Построим массив 28x28 заполненный единицами

```
In 2 1 n = 28
      2 a = np.ones([28,28])
```

Заполним диагональ массива единицами, а выше диагонали нулями

Рисунок 4 – Задание 1.4

Задание 1.5

Вывести свойства матрицы изображения на экран.

```
In 1 1 import ...
```

```
In 2 1 img = cv2.imread('Cat.jpg', 0)
      2 cv2.imshow('image', img)
```

Вывод класса:

```
In 3 1 print(type(img))
```

```
<class 'numpy.ndarray'>
```

Кортеж числа строк и столбцов (разрешение)

Рисунок 5 – Задание 1.5

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 2 1 import ...
```

Определение размера матрицы

```
In 3 1 img = cv2.imread('Cat.jpg')
      2 print(img.shape)
      3 cv2.waitKey(0);
```

```
(1199, 1200, 3)
```

Рисунок 6 – Задание 1.6

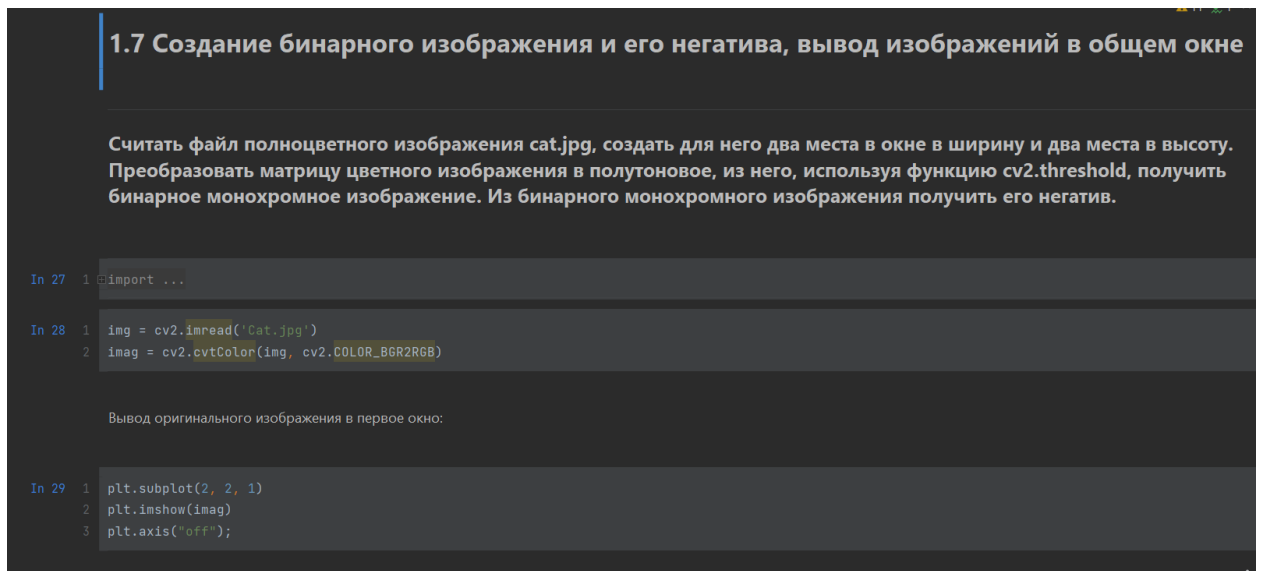


Рисунок 7 – Задание 1.7

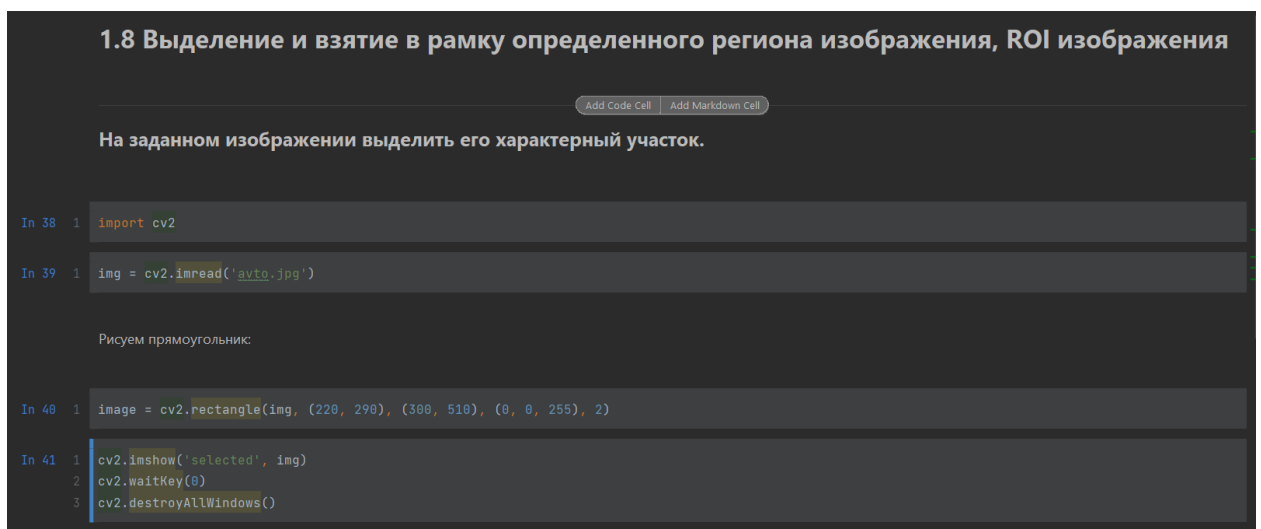


Рисунок 8 – Задание 1.8

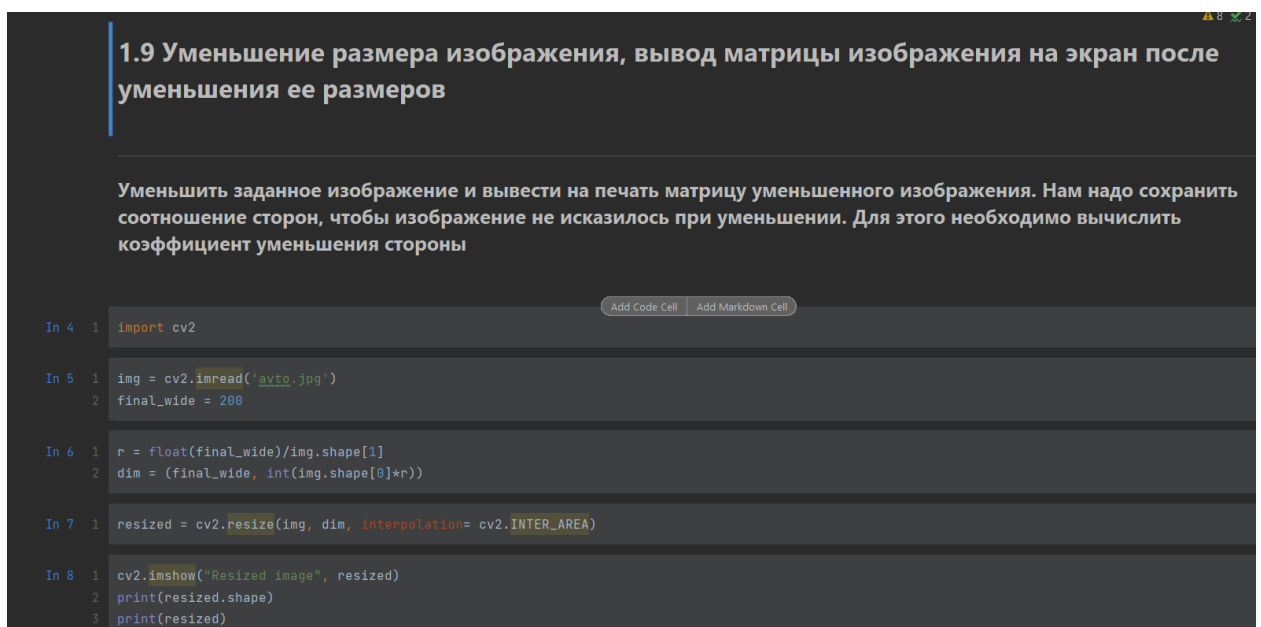


Рисунок 9 – Задание 1.9

1.10

Считать цветное изображение, конвертировать его в полутоновое, затем инвертировать изображение.

```
In 2 1 import cv2

In 3 1 img = cv2.imread("avto.jpg", 0)
    2 img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

Функция инвертирования изображения

In 4 1 img = cv2.bitwise_not(img)

In 5 1 cv2.imshow('Img', img);
    2 cv2.waitKey(0)

Out 5 -1
```

Рисунок 10 – Задание 1.10

Индивидуальное задание

```
In 37 1 import ...

In 38 1 image_gray = cv2.imread('Ind.jpg', 0)
    2 image_color = cv2.imread('Ind.jpg', 1)

In 39 1 im_bw = cv2.threshold(image_gray, 128, 255, cv2.THRESH_BINARY)[1]
    2 im_bwa = cv2.threshold(image_color, 128, 255, cv2.THRESH_BINARY)[1]
    3 im_bwb = cv2.threshold(image_gray, 128, 255, cv2.THRESH_BINARY_INV)[1]

In 40 1 plt.subplot(2, 3, 1)
    2 plt.imshow(image_gray)
    3 plt.axis("off")
    4 plt.subplot(2, 3, 2)
    5 plt.imshow(image_color)
    6 plt.axis("off")
    7 plt.subplot(2, 3, 3)
    8 plt.imshow(im_bw)
    9 plt.axis("off")
   10 plt.subplot(2, 3, 4)
   11 plt.imshow(im_bwa)
   12 plt.axis("off")
   13 plt.subplot(2, 3, 5)
```

Рисунок 11 – Индивидуальное задание