МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Костромской государственный университет» (КГУ)

Институт автоматизированных систем и технологий

Кафедра информационных систем и технологий

Направление подготовки/Специальность\* 09.03.02

Информационные системы и технологии

Профиль Разработка программного обеспечения информационных систем

Дисциплина Технологии компьютерного зрения

ОТЧЁТ

Лабораторная работа №0: Загрузка и сохранение изображений.

Выполнили студенты Смирнов Кирилл Андреевич

Шкунов Владимир Викторович

Группа 22-ИСбо-2а

Проверил Орлов Александр Валерьевич

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_

Кострома 2024

**Задание**

Задание 1: Создайте приложение на Python, которое загружает изображение из заданного файла и выводит в консоль его размер в формате "столбцы x строки".

Для загрузки изображения используйте функцию **cv2.imread()**. Укажите желаемую цветовую систему как **cv2.IMREAD\_COLOR**.

Проверьте работу программы на файлах с именами в ascii (латиницей) и с кириллицей.

Примечание: возможны проблемы с загрузкой файлов с не-ascii именами. Это не является основанием для незачёта работы.

Задание 2: Модифицируйте вашу программу так, чтобы загрузка изображения происходила в 2 этапа:

1. загрузка заданного файла как последовательности байт в массив numpy с помощью функции **numpy.fromfile()**.
2. разбор этой последовательности байт как изображения с помощью функции **cv2.imdecode()**. Укажите желаемую цветовую систему как **cv2.IMREAD\_COLOR**.

Выделите код загрузки в отдельную функцию, принимающую строку с путём к изображению и возвращающую объект изображения (массив numpy). Проверьте работу программы на файлах с именами в ascii (латиницей) и с кириллицей.

Задание 3: Модифицируйте вашу программу, чтобы после загрузки изображения оно было показано на экране с помощью функции **cv2.imshow()**. Чтобы задержать выполнение программы до нажатия клавиши, используйте функцию **cv2.waitKey()**. Имя окна должно совпадать с именем загруженного файла.

Задание 4: Модифицируйте вашу программу следующим образом:

1. после загрузки изображения разделите его на три канала (0 - синий, 1 - зеленый, 2 - красный).
2. каждый канал выведите в отдельном окне с пометкой BLUE/GREEN/RED.
3. закодируйте каждый канал в содержимое файла формата ".png" с помощью функции cv2.encode() и запишите его в отдельный файл.

Дополнительное задание: Используя функцию **cv2.selectROI()**, напишите программу, которая позволяет пользователю выделить прямоугольную область на изображении. Эта область должна быть сохранена в отдельный файл, имя которого вводится с клавиатуры.

**Вопросы**

1. Какой способ представления изображения был использован при выполнении работы?
2. Как при использовании массивов numpy узнать их размер по той или иной оси?
3. Как создать пустой массив numpy размером Y строк и X столбцов, элементы которого являются байтами (0..255)?

**Вопрос №1**

Мы использовали способ трехмерного индексирования или трехмерная дискретизация. В этом подходе изображение представляется в виде трехмерного массива (высота, ширина, количество каналов), где каждый элемент массива содержит значение пикселей изображения в формате BGR. Например, для доступа к синему каналу пикселя мы сможем обратиться image[i,j,0]. В 4 задании таким способом мы смогли вывести каждый канал в отдельном окне.

**Вопрос №2**

Чтобы узнать размеры массива NumPy, можно использовать атрибут share, возвращает кортеж, в котором содержится размеры массива по каждой из осей. share[0] вернет размер по оси Y, а share[1] вернет размер по оси X.

**Вопрос №2**

Чтобы создать пустой массив, можно воспользоваться функцией numpy.zeros. Например, empty\_arr = numpy.zeros((Y,X), dtype=numpy.uint8), где X и Y количество столбцов и строк.

**Скриншоты работы**

**Задание 1**

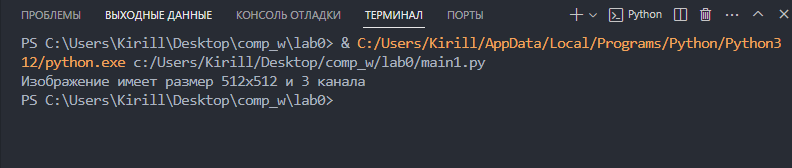


Рис. 1Вывод первого задания.

**Задание 2**

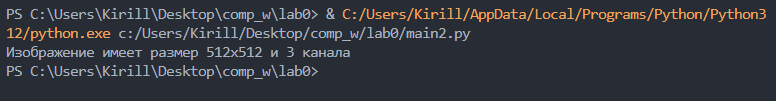
****

Рис. 2 Вывод второго задания.

**Задание 3**

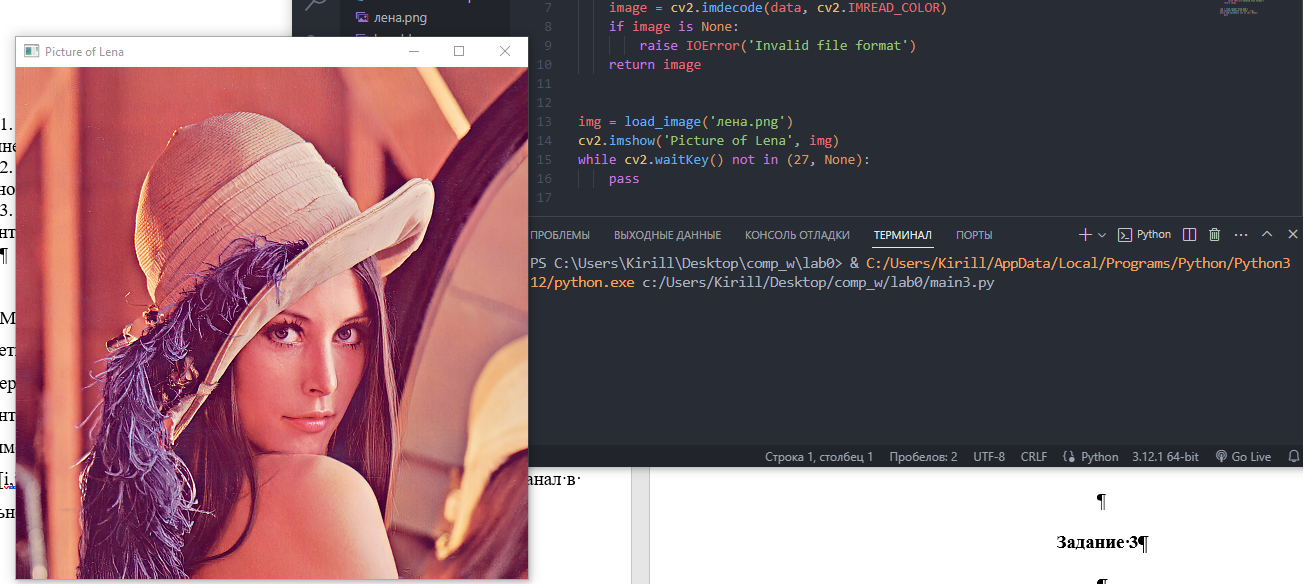
****

Рис. 3 Вывод третьего задания.

**Задание 4**

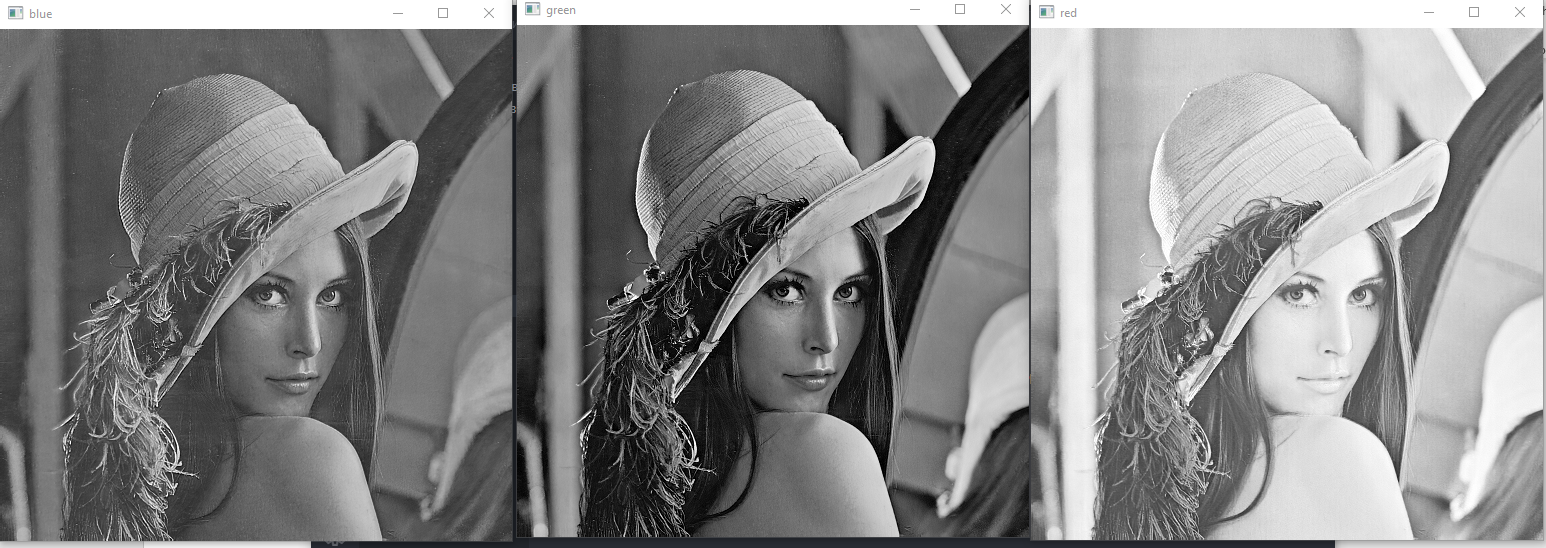


Рис. 4 Вывод четвертого задания.

**Дополнительно задание**

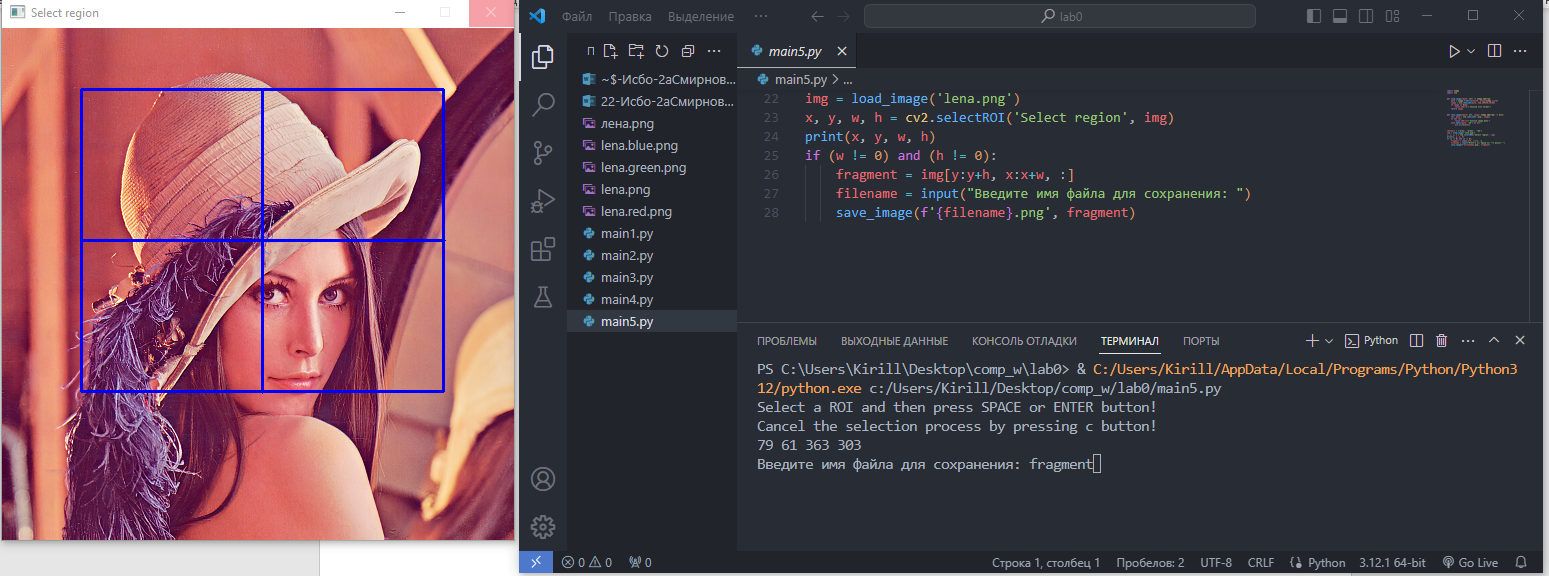
****

Рис. 5 Вывод дополнительного задания.

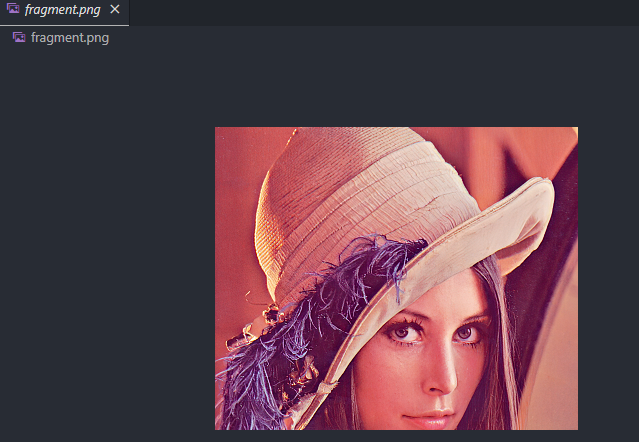


Рис. 5 Сохраненный фрагмент.